Clustering neuronal activity from fluorescent simulated traces and fluorescence calcium images by inferring spicular neuron activity

Jose Alejandro Cordero Rama

September, 2013

A thesis in the Master of Research on Information and Communications Technologies in partial fulfillment of the requirements for the degree of Master of Science at Universitat Politècnica de Catalunya. Barcelona Tech

Thesis directors:

José Adrián Rodríguez Fonollosa

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Asanobu KITAMOTO

Abstract

This project, developed in the National Institute of Informatics (NII, Tokyo) in collaboration with the National Institute of Genetics (NIG, Mishima), reviews some of the existing bibliography related with functional Multineuron Calcium Imaging (fMCI) and describes the implementation of a framework to analyze fMCI with the final aim of inferring the connection between the different recorded neurons by analyzing the activity patterns related which each one of these neurons.

The described framework gives a structure to develop some common algorithms in the bibliography including algorithms to generate synthetic signals to be used as a test set to the developed algorithms.

We tune and test our algorithms with synthetic data and after that we check the potential of our methods with real observations of zebrafish cortical habenula. We propose some improvements in the already existing algorithms found in the bibliography to clusterize the activity of the processed neurons, to find the underlying spicular activity and to generate synthetic data which fits properly with the real data.

This project starts with the review of the bibliography related with fMCI analysis techniques and the techniques which are commonly used in its analysis. After this review, an analysis of the available datasets recorded by the NIG is done and we propose some ways to fit this real data with the models commonly used by Neuroscientists when dealing with this kind of signals. After the discussion of the state of the art and after the analysis of how this existing proposals can be used with our real data and the existing limitations, we define a pipeline and split it in blocks; then we proceed with the detailed explanation of the implementation made for each proposed block and the discussion of how some implemented modifications help these algorithms to work properly with our real datasets. This framework includes tools to complete the task for each one of the proposed blocks: tools to generate synthetic signals, compute objective quality metrics over these synthetic signals and tools to use the developed tools with real data. The project finishes testing the implemented algorithms in real data and discussing the results and the potential of the methods to be used with real bigger datasets.

This framework is expected to be improved in the future by the NII and the collaboration of the NIG analyzing longer datasets and tunning the proposed methods to extract biological meaningful information of these recordings.

Acknowledgments

The development of this project has given me not only an academical and professional experience but the chance to experience in first person what means living in a different country right in the other side of the planet. I wouldn't have had this opportunity without the support of the NII Internship programs which supports these some-months internships by helping their interns economically.

I want to thank Professor Asanobu KITAMOTO for his efforts to make their students feel comfortable in Tokyo and helping us in our cultural adaptation. Many thanks to my colleagues and friends in Tokyo for making of this experience something unforgivable.

My biggest thanks and gratitude go to my advisor Professor José Adrián Rodríguez Fonollosa for his patience and his support during these last months.

Finally, I would like to thank my family for their love and support in every aspect of my life.

Contents

Contents **2**

1 Introduction **4**
1.1 Understanding how the brain works . . . 4
1.2 Functional multi-neuron calcium imaging (fMCI) . . . 6
1.3 Target dataset and technical considerations . . . 9
1.4 Objectives . . . 12
1.5 Master Thesis contribution . . . 13
1.6 Structure of the report. . . . 14

2 Extraction of neuronal activity from fMCI **15**
2.1 Watershed segmentation . . . 16
2.2 Dictionary based . . . 19

3 Neuronal activity processing techniques **20**
3.1 fMCI temporal series analysis . . . 20
3.1.1 Preprocessing . . . 21
3.1.2 Characterizing . . . 21
3.1.3 Spikes location techniques . . . 22
3.1.3.1 Statistical Modeling Approach . . . 22
3.1.3.2 Unsupervised adapted filter approach . . . 25
3.1.3.3 Shape adapted approach . . . 26
3.2 Finding functional clusters . . . 29

4 Proposals and implementation **37**
4.1 General Scheme . . . 37
4.2 Pipeline structure . . . 39
4.2.1 Signal Loader . . . 39
4.2.1.1 Data loader from fMCI . . . 42
4.2.1.2 Synthetic data loader . . . 43
4.2.2 Event Detector . . . 45
4.2.3 Event detector results tester . . . 50
4.2.4 Signal clustering . . . 52

4.2.5 Clustering results tester . . . 54
4.2.6 Results plotter . . . 54

5 Results **56**
5.1 Results using synthetic data . . . 57
5.1.1 Dataset generation . . . 57
5.1.2 Systems tested . . . 58
5.1.3 Results format . . . 58
5.1.4 Testing . . . 59
5.1.4.1 Clustering algorithms testing: . . . 59
5.1.4.2 Impact of spike detection performance in clustering performance . . . 60
5.2 Results using real data . . . 65

6 Conclusions **69**

Bibliography **70**

Chapter 1

Introduction

This project has been developed in collaboration with the National Institute of Genetics (NIG, Mishima) which is working in some projects related with the understanding of how the brain works. Since Ramon y Cajal discovered that the brain is a rich and dense network of neurons [29, 30], neuroscientists have been intensely curious about the details of these networks, which are believed to be the biological substrate for memory, cognition and perception. While we have learned a great deal in the last century about macro-circuits (the connectivity between coarsely-defined brain areas), a number of key question remain open about "micro-circuit" structure, i.e., the connectivity within populations of neurons at a fine-grained cellular level.

The objective of the NIG is to continue studying the details of these "micro-circuit" structures directly related with the activity of individual neurons which currently is poorly defined.

In this chapter we are going to briefly describe the story and evolution of these studies and the techniques used to record the material we are going to analyze. We will continue explaining the database we have and what are the objectives and the contribution of this project.

1.1 Understanding how the brain works

Two complementary strategies for investigating physical micro-circuits have been pursued extensively.

Anatomical approaches to inferring circuitry do not rely on observing neural activity but the physical or anatomical structure of the connections between neuronal structures. Some of these techniques are array tomography and serial electron microscopy [38]. This is not the strategy we are interested in, but it's important that these techniques exists because the final objective of the NIG is to check the results obtained with the techniques exposed in this project (and in future projects) with an anatomical approach.

The strategy we will work on is the analysis of the electrical activity of each one of these neurons. The brain and, more generally, nerve tissue if made of interconnected neurons; these connexions are called synapses and the connection itself is based on the transmission of electrical signals. Inferring these connections by observing only the electrical activity of these neurons allows biologists to study not the physical interconnection of these neurons but functional structures and what is more, these

techniques allow "in vivo" experiments, which means that these neural activity can be observed and studied under different environment or applying different stimulus to the subject under study to find if there are specific connections which are only active under specific circumstances.

The essence of cortical function is the propagation and transformation of neuronal activity by the cortical circuit. How activity can propagate through a network composed of weak, stochastic, and depressing synapses is, however, poorly understood. It has been proposed that sequences of synchronous activity propagate through the cortical circuit with high temporal fidelity. Synchronous summation of excitatory post-synaptic potentials could ensure post-synaptic firing and the nonlinear gain by the spike threshold could preserve temporal fidelity so reactivations of the same chain would result in exact repetitions of precise firing patterns. Repetitions of temporally precise firing sequences of spikes have been reported although their existence is controversial [37].

Although there is some controversy related with the existence of these firing patterns, we assume this existence as part of our hypothesis and we will define our goals and the development of this project assuming that these firing patterns exist and we will set that our goal is to provide NIG scientists tools to find these patterns and evaluate them. In future, using approaches similar to the one we will explain in this document or in parallel researches, biologists should definitively accept or reject the existence of these patterns we will assume.

What seems clear and there's not controversy about is the behavior of a single nerve cell. The electrical activity of these cells happens because of the propagation of calcium ions Ca^{2+} through the nerve cell membrane. This mechanism is very common in biological beings to generate electrical stimulus. The first idea we present here is the use of electrodes to record this electrical activity.

The electrodes used in electro-cardiography (ECG) or electro-encephalography (EEG) record a mix of signals in a non-invasive way and they allow the recording of in vivo activity. The problem of these techniques is that they don't allow us to analyze the behavior of a specific structure, but the mix of the signals generated by several structures.

A similar technique can be used to examine the behavior of smaller zones (for example, by inserting electrodes inside the tissue in a quirurgical approach) but they can't avoid the problem of recording the mix of several signals.

Direct electrical measurement using extracellular electrode arrays and spike-sorting algorithms are routinely used to examine neuronal activity in vivo. Such techniques are very powerful; they allow the simultaneous recording of small groups of neurons, typically up to 25 in insects and 100 i rodents, with an excellent temporal resolution ($< 10\mu s$); yet, precise identification of the recorder neurons usually remains difficult to impossible. Although these techniques are quite specific, they have the same problem than EEG or ECG and is the difficulty to find exactly what structure is generating the observed activity. Independent Component Analysis (ICA) and similar algorithms allow us to identify the activity of the original sources, but we can't still assign each one of them to a specific neuron.

In Figure 1.1 an example of how these kind of signals look like is shown; this signals has been recorded with a sampling rate of 15Khz. The activity looks quite spicular when recorded at 15Khz, which means that the movement of ions throw the membrane is very fast. Although the spiking behavior of the analyzed signals, the exact instants when these spikes happens can be perfectly located, so this family of techniques are considered to provide a very good temporal resolution.

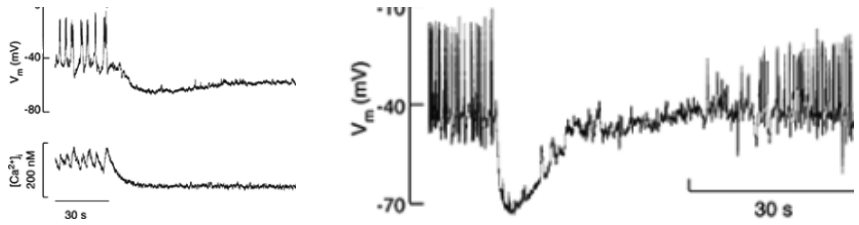


Figure 1.1: Spike activity extracted from an experiment recorded using extracellular electrode arrays and a sampling rate of 15kHz.

These family of techniques has been widely used since now, but nowadays the objective is not just analyze how the brain reacts as a whole structure, but find the "micro-circuits" and how they are connected. This means that we need other techniques capable to identify not only the activity of a structure, but the behavior of the parts which are part of that structure (in our case, neurons).

Optical techniques such as calcium imaging [21] have also been used in vivo with some success in insects and vertebrates. In this case the advantages are more or less reversed: cell identification is easy, but estimating spiking activity from neuronal cytoplasmic calcium variation is difficult, especially when single spike resolution is desired.

The NIG is very interested in analyzing these kind of material and in this project we will work on techniques to analyze one of these optical techniques called functional Multi-neuron Calcium Images (fMCI).

1.2 Functional multi-neuron calcium imaging (fMCI)

Functional multi-neuron calcium imaging (fMCI) can record huge populations of neurons with single-cell resolution [24, 25, 7]. In other words, fMCI can reconstruct when, where and how individual neurons are activated in a network of interest, although its strategy is somewhat invasive to living biosystems. For fMCI brain tissue is bulk-loaded with calcium-sensitive fluorescence indicator and the changes in fluorescence intensity are measured from the cell bodies of neurons.

As we explained in the previous point, the movement of calcium ions through the membrane of neurons is how biological currents are generated and propagated. The calcium-sensitive fluorescence introduced in the nerve cells is excited and its bright is recorded with a microscope. One of the most extended technique is known as two-photon imaging. An scheme of this technique is shown in Figure 1.2.-

The output of this technique is a video where the activity of the neurons appear as a change in the intensity or bright of the pixels which correspond to the active neuron. As we discussed before, this techniques allows us to know exactly the position and the activity of hundreds of neurons at the same time. There are two facts which makes this technique very useful to study the relation in the firing activity of neurons and brain structures: if offers a very good spatial resolution and it offers the synchronous observation of thousand of neurons.

Despite everything, this technique is not perfect; this technique has a very poor temporal resolution and this problem has not already been solved. There are two main reasons for this:

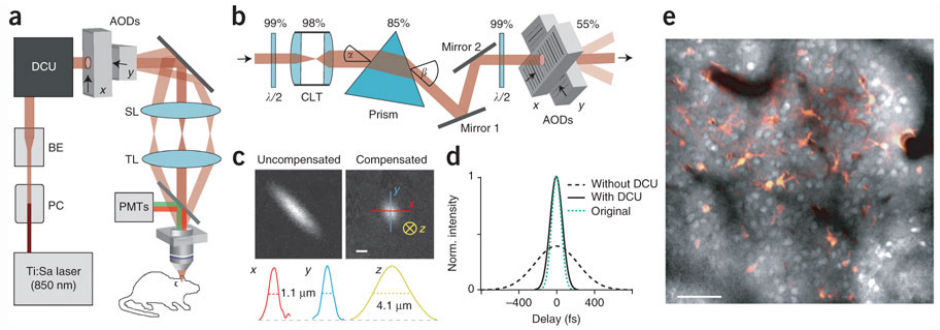


Figure 1.2: General scheme of two-photon imaging of neocortical neurons in vivo. (a) Schematic of the two photon microscope setup, (b,c) setup of the lens of the microscope and technical details, (c,d) differences dependent on the laser receiver setup, (e) example of a two-photon image of a cell population in mouse neocortex. This image only try to help the reader to make a general idea of the process, more details can be found in [5].

- Microscopes usually has very high resolution, but is difficult to record videos with high frame rate.

  As we shown in the previous point of this report, neuronal electrical activity is seen as spikes when the sample rate is about 15Khz. fMCI techniques usually record the videos with a frame rate of 5-30 frames/second (i.e. 5-30 Hz), so the spikes can easily be lost.

- The fluorescence reaction of the calcium indicator is not immediate.

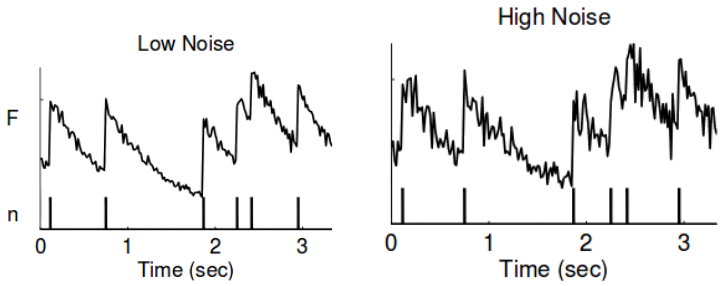
  Once the indicator has been excited, the part of the image which corresponds to that neuron gets brighter but it take some time to the indicator to lose its bright and become dark again. This darken process describe and exponential shape if this bright is observed over time. The general scheme of what happens in the process is that an spike is generated by the activation of the neuron, this suddenly increase the calcium in the nerve cell and it reacts with the indicator which makes the nerve cell brighter. The bright of the nerve cell decrease in time exponentially with a decay about an order of magnitude slower than the time of the underlying neural activity ([5, 24]).

fMCI technique has not been discarded as a useful tool even when we can't directly see when the neuronal activity is happening due to the low frame-rate because a neuron needs some time to lose its bright once it has been activated; so we can actually know that a neuron has been activated although we may not know exactly when this has happened.

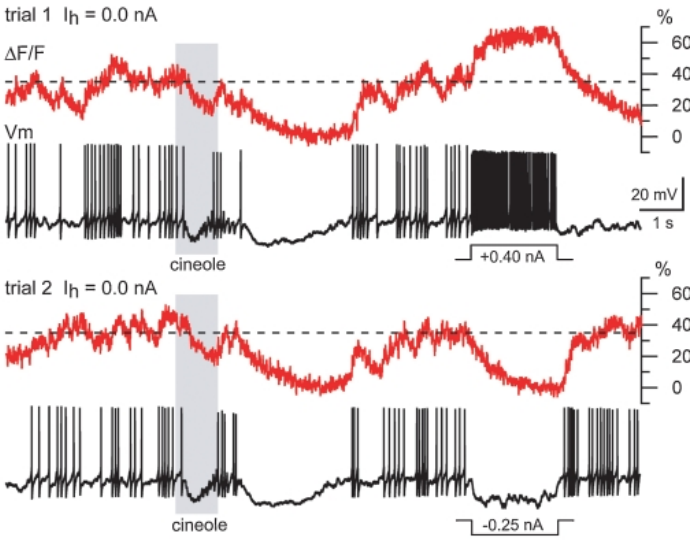
Exist lots of papers which discuss different algorithms to find where the spikes has actually happened just by analyzing the bright of each nerve cell. These papers usually validate the behavior of their algorithms by recording at the same time the desired nerve cells using fMCI techniques and multi-electrode arrays techniques, so they can try to recover the spikes detected by the multi-electrode array techniques just by analyzing the videos recorded using fMCI.

In the Figure 1.3 we can see the previously described characteristic shape of the bright caused by the activation of a nerve cell and the synchronous recording of the two previously mentioned techniques.

This techniques allows us to know exactly the position of the observed neurons and allows us to

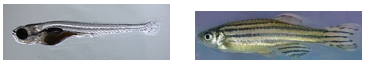


(a) Two simulated fluorescent data generated by the same spike pattern following the model proposed by [33].



(b) Coupled signals recorded by fMCI and multi-electrode arrays. There are some effects in the fluorescence signals we will discuss in following points of this project which hinder the task of finding the exact timing where the spikes happen. As we can see in this plot taken from [21] the exponential decay can be observed where the fluorescence signal is not much saturated and this task becomes almost impossible where the high spiking rate makes the fluorescence signal to reach its saturation level.

Figure 1.3



(a) (b)

Figure 1.4: (a) younger zebrafish, (b) Adult zebrafish

identify non-active nerve cells. The weakness of this algorithm is that finding when the electrical activity which originates the observed activity becomes an extra problem which must be solved.

1.3 Target dataset and technical considerations

This project has been developed with a group of biologists of the NIG (National Institute of Genetics, Mishima) which are working with zebrafish. Zebrafish, mice and e-coli are some of the animals typically used by biologists because of their life-time and the big amount of descendants they have in every generation. These allows biologists to analyze lots of generations in short periods of time. For fMCI the use of mice or zebrafish is commonly used because they are vertebrates but furthermore, zebrafish (due to its thin and transparent skin) allows the recording of fMCI without quirurgical intervention so it's easier for scientists to experiment with these animals using different stimulus in vivo and even without anesthesia.

The zebrafish are tropical freshwater fishes which were originally found in slow streams, rice fields and the river in East India and Burma. It has horizontal blue tripes on the side of its body which extend to the end of the caudal fin as shown in figure 1.4. Their shape are spindle-shaped and compressed at lateral with their mouth upwards. Zebrafish eat smaller living organisms than themselves such as plankton, insect larvae, worms and small crustaceans. They are popular aquarium fish and important vertebrate model organism in scientific research [39].

The model of zebrafish organism is useful for studying the development of vertebrate and gene function which can encourage the research in higher vertebrate models such as rats, mice and human [39]. Because of the zebrafish's eggs are clear and their development happen outside of the mother's body, it can assist the scientists to observe a zebrafish egg grow into embryo until to be newly formed fish via a microscope. The scientists perceive the cells divide and form different parts of the fish's body such as eyes, heart, liver, stomach, skin, fins, etc. Until the fish is complete its organ. Consequently they have a significant role for understanding the cell development and helps on the scientists with discovering the causes of birth defects in human and overcome this problem in future. The study of zebrafish in this research is focus on the detection of nerve cell signal in the video.

In the first meeting with NIG biologists they told us that a new microscope with high resolution imaging capabilities had been bought recently. We arranged that during the internship, they would record some fMCI videos of zebrafish, probably (although they told us that it would be difficult for them to record that way) with information related with the stimuli applied to the fish while it was being recorded.

As a first material to start our project, they provide us with some fMCI material they recorded with their old microscope. Although in that material were some videos, they were only interested in one of them, the recording of a section of the head of zebrafish. They are very interested on study the

neuronal activity under visual stimulus, so the 'head' of the fish is recorded. In Figure 1.5 4 samples randomly selected from the only video recorded from the zone of interest.

We started working in general techniques to work with these kind of material while we waited for new material to check our results, but NIG argued that they had some technical problems with their new microscope. The consequence of this technical problems has been that we have not had real data to test our results during the development of this project.

Although we have been working in general models and we have not had the opportunity of checking them with real data as desired, we have discussed a lot with the NIG about the properties expected of the analyzed signals to adapt our algorithms and techniques as much as possible to the real data they will work with when the technical problems with their new microscope are solved.

The main conclusions and specifications extracted from the meetings with NIG can be resumed in the following points:

- New microscope data won't have much higher frame rate than the available dataset.

  We will assume a slow frame rate of 10Hz (the same than the available dataset) understanding that this is the worst case, if the final videos have a higher frame rate that would never be a problem. That higher frame-rate should mean that the analysis of the extracted signals should be easier. Even if somehow that's not the case, an appropriate down-sample should make these new videos compatible with the tools and algorithms developed in this project.

- New microscope data will have more resolution than the available dataset.

  This information allows us to be very critical with the results in the image processing step. We should not discard an algorithm directly if the results are not as good as we expect. Before discarding anything we have to analyze the reasons of these hypothetically not-so-good results and evaluate if they could be better with higher resolution images.

- Nerve cells activity can be very dependent on the characteristics of the nerve cell itself.

  Factors like the size of the neuron or the saturation offset of calcium (resulting in a fixed minimum bright of the neuron) can generate shapes where the exponential behavior due to the fluorescence can be difficult to distinguish. It's important to take into account that fMCI techniques is a non-linear function related with the spicular or electrical activity of the nerve cells which means that this minimum bright on a nerve cell is not only an offset which can be removed by a simple low-pass filter, but a phenomenon which modifies the expected shape of the signal due to the saturation intrinsic to this techniques. Some examples can be shown in 1.6.

- Sparsity assumption in the activity of nerve cells and non-single spike activity pattern.

  - Sparsity assumption in the activity of nerve cells

    Neurons in the zone of interest of zebrafish stay most of the time inactive during long observations. This allows us to assume a similar sparsity assumption than the one made in papers like [38, 5, 10] and others.

  - Non single-spike activity pattern.

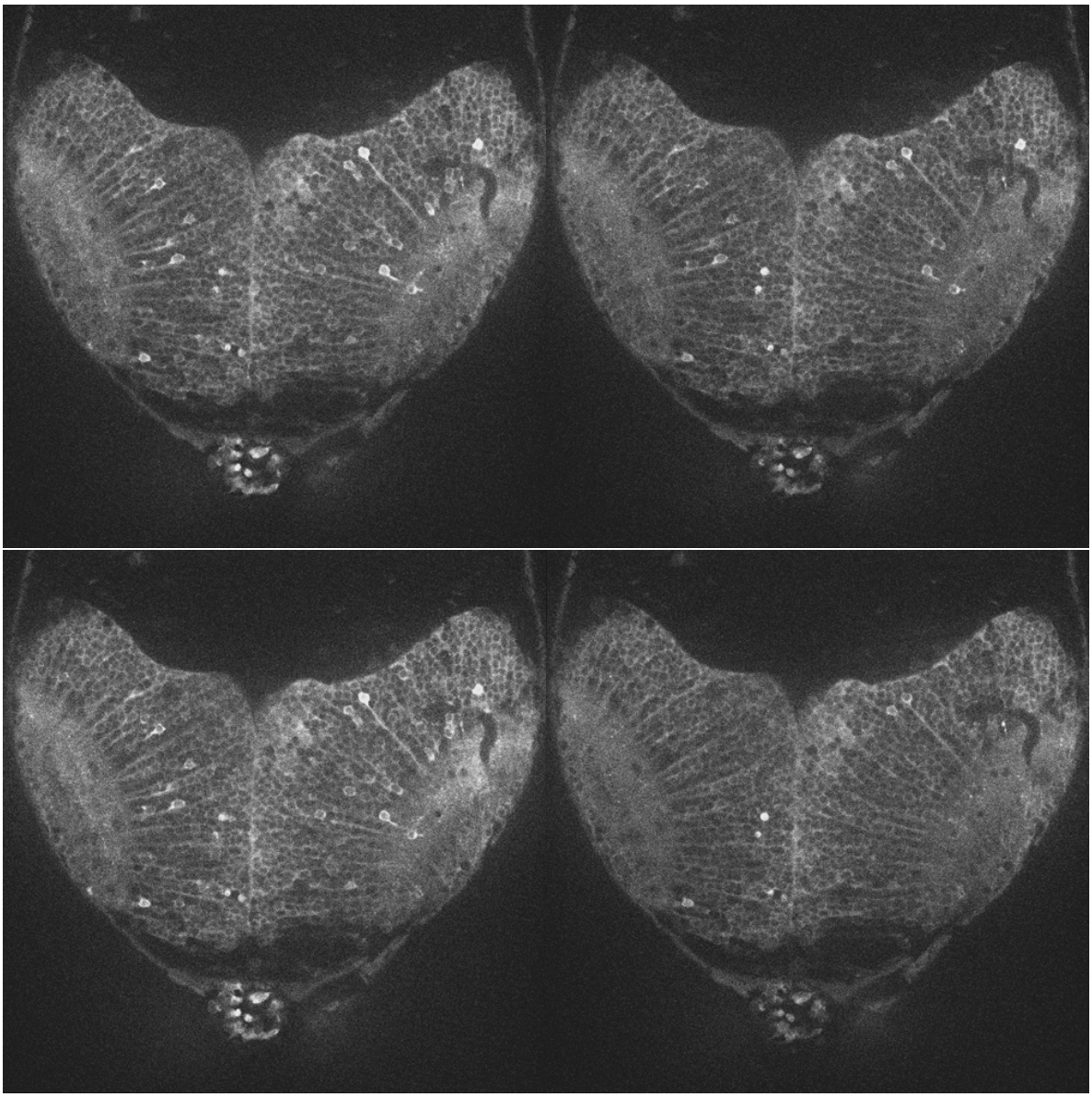
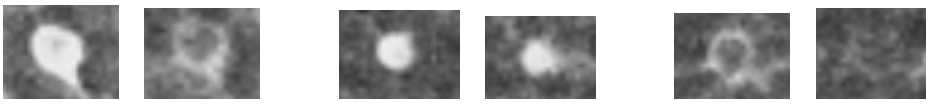


Figure 1.5: Four frames of the available dataset.



(a) (b) (c)

Figure 1.6: Three different neurons with different shapes and concentration of calcium when not firing. (a) shows a nice behavior which generates easily detectable exponential shapes. (b) shows a nerve cell which is always highly saturated with calcium, the increase of bright when the neuron is activated can be distinguished, but is not so easy. (c) shows a nerve cell which present an activation pattern different from the two previous ones, in this case only the border of the neuron gets brighter when it's activated.

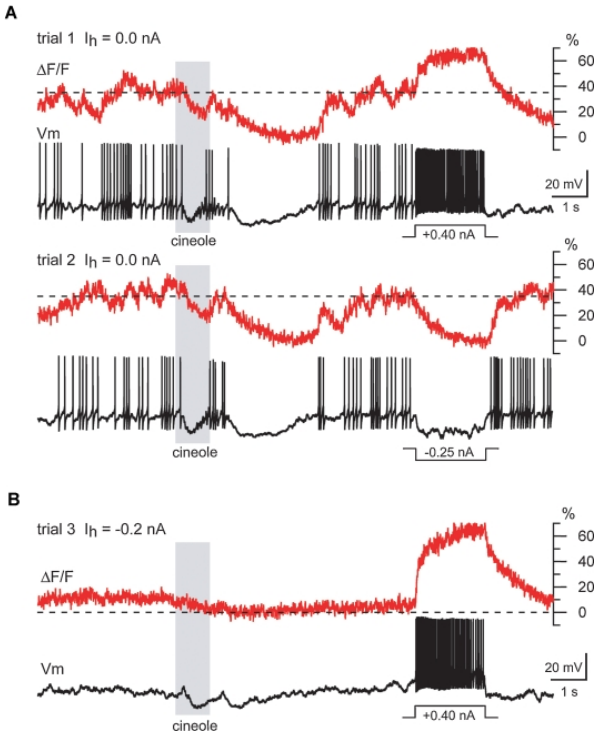


Figure 1.7: Reaction of fluorescent calcium imaging techniques under different electrical stimuli shown in [21].

Although the sparsity assumption previously discussed, NIG biologists warned us that the assumption made in some technical papers does not correspond at all with the real phenomenon observed in zebrafish. They are not interested in recording simultaneously the neuronal activity of zebrafish using multi-electrode arrays and fMCI but they did it in the past and they assure that zebrafish nerve cells has a behavior similar to (B) in Figure 1.7, i.e., bursts of spikes happening in short periods of time.

Taking into account these two considerations, we must assume that we won't find activity patterns as the ones shown in Figure 1.7 (A) , but something similar to the spike pattern shown in Figure 1.7 (B). There are some saturation considerations which makes the exponential decay in the calcium imaging difficult to appreciate (Figure 1.7 (A)) but we won't focus our techniques in solving this problem since it's not expected to appear in the videos recorded by NIG.

1.4 Objectives

The first objective set for this project was to check the existing bibliography and try to adapt it to the material recorded by the NIG with the objective of finding the underlying functional relation between different neurons by observing their activity patterns using fMCI techniques.

Since the NIG has had some technical problems which have made impossible to get new material we have been forced to modify both the objectives and the scheduling of the project whilst we were

developing it.

The fact of having only one video of zebrafish's region of interest and being this video only 120 frames long limits a lot the possibilities we have. First of all, the whole sample is recorded at 10Hz, which means that the available observation's duration is about 12 seconds.

The impact that one of the specifications given by the NIG is very critical in the analyzed data: most of the time the neurons are not active. This means that most of the neurons in the available video are non-active during the hole sequence or they fire at most one or two times.

Developing algorithms to work with these datasets is very difficult because we don't have any labels to check if our algorithms work well or not and because even if we decide to use our intuition to check it the data is too restricted to only one fish (we can't assure that our system can generalize to other fishes) and to a very short signal (the relations we will find may be just coincidence and not the consequence of a biological meaningful connection).

This makes really difficult to adapt our algorithms to this data specially if we look at the approach which the NIG wants us to follow, an approach based exclusively on the observations.

Taking this into account, we have set some secondary objectives whose objective is to propose several options to the future intern students of the NIG to choose the most promising ones taking into account the characteristics of the signals the NIG will provide when their technical problems are solved and longer datasets are available.

- Main goal:

  Develop some tools for the NIG to analyze their fMCI images with the objective of helping them to find the underlying functional relation between different neurons of zebrafishes.

- Secondary goals:

  - Propose different alternatives based on the assumptions and specifications suggested by the NIG biologists.
  - Make the proposals flexible enough to be adapted when new data is recorded and available.
  - Since the data does not seem enough to train reasonably any statistical system unless we set a lot of restrictions (which is not desired), propose some system to quantify the behavior of the developed algorithms to evaluate their potential.

1.5 Master Thesis contribution

This Master Thesis makes a review of the existing bibliography and analyzes the problem both in an empirical way based on real data and in a theoretical way trying to avoid the lack of available material. We propose some implementations and modifications of published systems to make them fit with real data and to improve their potential.

The implementations detailed in this project covers the whole fMCI analysis process:

- Image processing for neuron detection
- Processing of individual neuron data and spike recovery

- Clustering and identification of functional structures

The project proposes, too, a framework to test all these algorithms with synthetic data generated to accomplish the specifications suggested by NIG biologists. This framework includes algorithms to generate synthetic data and to measure the quality of the proposed systems and algorithms based on this synthetic data.

Finally, some improvements has been proposed as alternatives of the methods proposed in the bibliography to adapt them to the real data we want to analyze. These improvements have been made specially for synthetic data generation and for clustering and identification of functional structures.

1.6 Structure of the report.

After this brief introduction we will present in the Chapters 2 and 3 the different techniques regarding to fMCI we have found in the bibliography. We split the image processing part from the signal processing part since this project and document is focused in the signal processing part.

Chapter 2 give a brief explanation of the general idea of how the image is processed to detect the location of the neurons and how a signal corresponding to the fluorescence activity due to the electrical activity of the signal is computed using the bright of the detected cell frame by frame.

Chapter 3 is focused on the signal processing part of this project. We will discuss some of the techniques which are used and we will discuss their strong and weak points.

In Chapter 4 we present the architecture of the system we have built as a framework to analyze these kind of videos. Some details of the implementation are given and some improvements or alternatives made to the existing algorithms in the state of the art are suggested. In this chapter there's not much discussion about the limitations of the algorithms because we consider that has been already explained in the previous chapter.

In Chapter 5 we will explain which experiments we consider necessaries and why and show the results of these experiments. After that we comment these results and try to extract some conclusions and draw the lines for future development of this framework by future researchers assigned to this project.

Finally, Chapter 6 summarizes the conclusions of the project.

Chapter 2

Extraction of neuronal activity from fMCI

The action of zebrafish nerve cell in the movie are displayed as blinking by brighter and darker. The nerve cells which are active become brighter and the nerve cells which are nonactive will be dark or stable. The final aim of studying the behavior of zebrafish nerve cells starts by using fMCI techniques starts detecting where the neurons are in the recorded videos and extracting the temporal sequence which correspond to the fluorescence of that neuron (i.e., a signal strongly related with the electrical activity of the neuron). The detection of zebrafish nerve cells by manual inspection is very time consuming and exhausting, the development of automatic nerve cell location and analysis is required.

In this point we are going to discuss some different alternatives to extract temporal series corresponding to the electrical activity of the neurons in the video.

The classical method for nerve cell detection is image segmentation. The basic method is image threshold [34, 13] however this technique is not able to identify the contiguous cells. Another technique which can be used in this problem is the watershed transform which can segment touching objects as long as separate initializing seeds can be found. The basic concept of watershed segmentation is based on the topographic representation of image intensity. Watershed segmentation also incorporate other principal image segmentation methods including discontinuity detection, thresholding and region processing. For this reason, watershed segmentation displays more effectiveness and stableness than other segmentation algorithms. Watershed methods use nuclei location information as seed of segmentation [14, 13]. A problem of watershed based segmentation is over-segmentation, due to noise and other irregularities of the gradient. This problem can overcome by marker-controlled watershed [1, 35].

Although these techniques are widely used, we will comment some other techniques based on the statistical analysis of the videos [11] using ICA based algorithms to find close pixels whose behavior is similar and defining those strongly correlated areas as neurons or Regions of Interest (ROI). This kind of algorithms won't be used in this project because when we have tested them we have had some problems detecting some neurons which are non uniformly activated; this algorithm works quite well with synthetic images generated as described in the paper, but they assume that all the pixels which are part of a neuron has quite a similar dynamics and this is not the case of the signals we are working

with.

In a similar line, [31] gives a method to find these cells in calcium imaging data but it has the same problem than [11]: its performance is very good with synthetic data and with their real datasets but the assumptions made are not applicable at all to the data analyzed in this project. Figure 2.1 summarizes the process followed in the cited paper which is a base for the literature related with automated analysis of calcium imaging data.

Finally, there's a last family of algorithms we have explored consisting on creating dictionaries which are used to train an automatic Machine Learning (ML) system to distinguish what is a neuron and what is not. This kind of algorithm require some supervision so it's not immediately applicable to the problem we want to solve but it will set a base for the implementation done which will be explained in the following points of the project. Although this algorithms require some supervision or previous knowledge to build the dictionary, it allows us to find not only neurons which are active in the observed period but non-active neurons based only in the shape of the neuron. We consider that this is quite important, because the lack of activity is a pattern which is important when we will work on the final part of the project: find functional clusters. This kind of approach would be specially interesting if the dataset contains information of the stimulus which are being used to excite the animal, so we could distinguish if appear some kind of functional cluster or connexions or not depending on the existence of a given stimulus.

The final implementation done in this project is a mix between Watershed Segmentation and Dictionary Based Neuron Location. Since this is not the main point of the project, we will explain something else in the next points but we won't give a very detailed description.

Both algorithms seem very adequate to the problem we are dealing with because they are capable to find both non-active neurons and active neurons and because they will potentially improve their performance with a higher resolution of the recorded videos (one of the points that NIG assure they will be able to do with their new microscope).

2.1 Watershed segmentation

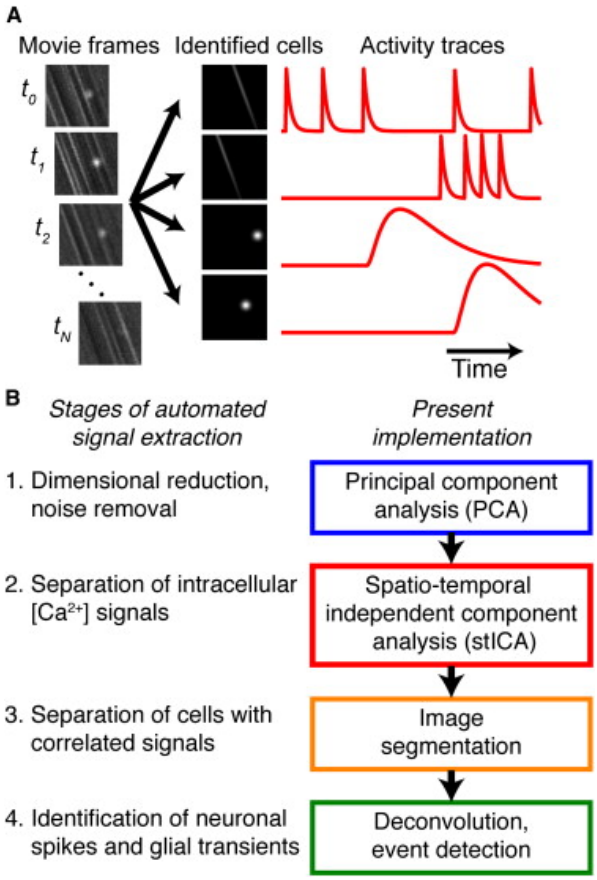
Watershed segmentation algorithm was developed in a previous internship by Pataraporn Promkumtan. We have used some modifications to adapt the proposed algorithm to our signals. In the following pages we will explain accurately the idea of the algorithm. This algorithm is explained carefully because it's the core of the algorithm we finally use to extract the temporal signals from the available videos.

Since we consider that with a basic idea of how the watershed algorithm works is enough to have a general idea of the potential and limitations of the algorithm, we present a basic idea of the algorithm and the results achieved by a previous student in another dataset.

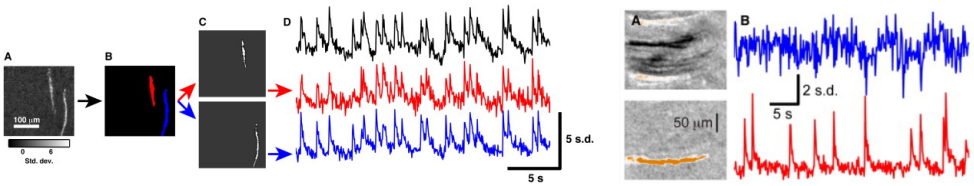
The concept of watershed is based on viewing the height of the area as proportional to gray intensity level of image as figure **??**. The bright areas have high altitude and dark areas have low altitude, then they might look like a geographic surface [23, 27]. In the geographic interpretation, we consider on three types of points:

a) The points from regional minimum of the surface

b) The points at which a drop of water would fall with certainty to a single minimum or the



(a) General scheme of the process. In A some movie frames, how active cells are identified and in red, the temporal series extracted from each detected neuron. In B the steps followed by the paper are enumerated and how each point is implemented is detailed. Steps 1,2 and 3 of this pipeline are under study in this point of the present report, being deconvolution event detection studied in the next point as a milestone by itself.



(b) Some examples of the statistical location of nerve cells and the related temporal series extracted from each one of them following the methodology explained in [11]. In the top image we can see in A the original frame, in B two detected nerve structures and in D their corresponding temporal sequence. The black sequence in D corresponds to the signal extracted from the entire frame A before finding their independent components. In the bottom image we can see how the same process allows not only to distinguish neurons in an image frame but to find a single nerve structure. One of the main problems of this algorithm and methodology is that only active nerve cells can be detected.

Figure 2.1: Explanatory plots to illustrate the procedure proposed in [11].

points at which construct a gradient interior region, are called as catchment basin or watershed of that minimum.

c) The points at which a drop of water would be equally fall to more than one minimum and form crest lines dividing different catchments basing, in termed by divide lines or watershed lines.

The objective of watershed segmentation is to find all of the watershed lines. Suppose that a hole is punched in each regional minimum and the entire geography is flooded by letting water leak at a uniform rate through the hole surface. If the water of neighboring catchment basing is likely to merge, a dam is created to avoid the water spilling from one basin into the other. A dam is built all the way to the highest surface altitude or to the maximum gray intensity value. Consequently only the top of dam represent above the water line. These dam boundaries are the divide lines of the watersheds. Therefore, the extracted edges of all dams form the watershed segmentation [23].

Dam construction is based on binary morphological dilation. At each step of the algorithm, the binary image is obtained as following. Initially, the set of pixels with minimum gray level 1, others 0. In each subsequent step, the water is flooded in the 3D topography from below. The pixels covered by the rising water are 1s and others 0s. The dam must be built to keep the water from spilling across the basins [30].

If at flooding step $n-1$ there are two connected components and afterward, at flooding step n, there is only one connected component; this indicates that the water between the two catchment basins has merged at flooding step n. Let q denote the single connected component. The dilation of the connected components in each step is performed by the structuring element and is subordinated to two conditions.

- Condition 1: The dilation has to be constrained to q. The center of the structuring element can be located only at the points of q during dilation or in the other meaning that it can not go beyond q during dilation.

- Condition 2: The dilation cannot be performed on the set of points that may cause the sets being dilated to merge.

When satisfaction by two conditions, the dam is constructed by the points on which the dilation would cause the sets being dilated to merge. Then, one-pixel thick connected path is build as separating dam at the n^{th} stage of flooding.

The watershed algorithm allows us to find both active and non-active neurons, which is a desirable property. They allow us to work with the images understanding them as a "map" so no external or prior information is required. The problem of this algorithm is that is very difficult to distinguish which of the 'flooded areas' are actually a neuron or are not. It has another problem, it's very sensitive to a good recording (high resolution) because it requires clearly separable neurons.

Anyway, as we discussed in the objectives of the project, the high resolution of the images should not a problem in the future because we expect future datasets to be much better than the currently available ones. The problem of distinguish whether a flooded area is or not a neuron becomes the main problem of this algorithm because we cannot expect any improvement even with better datasets.

2.2 Dictionary based

Searching neurons in the analyzed images by using a dictionary is inspired in the fact that in the images we are analyzing the neurons can be identified as a neurons by its shape as shown in Figure 1.5. As we introduced when we firstly commented this method, the main problem is we must build a dictionary.

In related bibliography [26, 32, 8] they have available a huge dataset and use some of the available dataset to define what kind of object is gonna be detected. Once this subset is selected, some techniques as k-Singular Value Decomposition (K-SVD) can be used to automatically learn the dictionary. This dictionary learning is commonly understood as objects in a subspace which can be combined linearly to recover the original object. Some more sophisticated techniques try to not only make a dictionary to identify the objects which are going to be detected, but with capacity to discriminate objects of different labels.

Once this dictionaries are calculated, the distance between the atoms of the dictionary and the unlabeled samples are computed and a label is assigned to these elements depending on that metric.

The problem in our case is that our dataset has only one real video of 120 frames. To use the same approach which is used in the cited papers presents two problems:

- Requires supervision and we want a unsupervised method.
- If we separate the video in two subvideos and we build a dictionary using one of these parts, if we generalize the method to other videos we are assuming that the neurons of different videos (i.e. different fishes and maybe not exactly the same zone) has common shapes.

In the implementation proposed point we will discuss a method to use this approach avoiding the two problems mentioned before.

Chapter 3

Neuronal activity processing techniques

Despite the tight link between chemical and electrical signaling in neural systems, experimental studies of neural information processing have mainly been limited to observations of the electrical activity of single neurons and small neural networks. Recently however, the new experimental technique known as two-photon calcium imaging has enabled the study of the concentration of calcium within the dendritic trees of single neurons as well as the time-varying calcium activity of neural populations in vivo. Studies of neuronal calcium activity have already confirmed the existence of functional maps in the visual cortex; however, in order to use this technique to study temporal correlations between distinct neurons, calcium imaging must also be able to show the precise time at which each cell emits a spike or action potential.

In this chapter we will take as an input the neuronal activity of each one of the neurons, assuming they have been successfully extracted from the video. What we will have as inputs are sets of coordinates (x, y) and fluorescence activity F, a positive magnitude defined, for each frame, as the mean value of the pixels of the neuron located in the position (x, y).

In this point of the process we will focus our efforts on how the bibliography suggests us to analyze this data. In this chapter we will explain the main techniques- The techniques implemented in this project will be detailed in the next chapter.

Once we will have discussed how to analyze individually each one of these signals, we will discuss how to analyze several signals. We will do it in order to accomplish the final goal of the project: finding functional structures and trying to suggest some patterns to the NIG scientists. We will do it by finding functional clusters. All the related considerations and assumptions are discussed in this chapter.

3.1 fMCI temporal series analysis

Two-photon calcium imaging is an emerging experimental technique that enables the study of information processing within neural circuits in vivo. As we explained in the introduction of the project,

1.2 the electrical activity of the neurons is very quick compared with the frame rate at which the videos have been recorded to. This activity looks like high frequency spicular noise even when they are recorded with high sample rates. Using fMCI techniques, with sample rate about 3-30Hz, these spikes are not detectable but spicular activity in the neurons generates a bright in the fluorescence signals with a very characteristic exponential decay which is observable even with this low sample rate.

Although the spatial resolution of this technique allows us to see the calcium activity of individual cells, inferring the precise times at which a neuron emits a spike is challenging because the spikes are hidden within noisy observations of the neuron's calcium activity or because some natural effects like several activations of a given neuron in a short period of time cause the overlapping of the evoked 'shapes' in the fluorescence signal by each one of the firing events and may be impossible to distinguish each one of these just by observing the final signals.

3.1.1 Preprocessing

The preprocessing of these kind of signals in the literature is done by a simple high-pass filter with cut frequency at 0.1-0.3 Hz. Although in several disciplines as ECG analysis the filtering techniques are very important, in this kind of signals the filter used isn't. This is because the properties of the signal which can be affected by the transients of the filter are not important at all to reach our final goal (in disciplines as ECG, a little inflexion in the shape of the signals can be very important for the diagnosis of a pathology).

In previous points we have talked about how to extract the signal related with a neuron by computing the mean of the pixels which are part of the neuron and we have called it "fluorescence signal". In the related literature they usually express this signal relative fluorescence changes or $\Delta F/F$. They compute this $\Delta F/F$ by subtracting the background of a given nerve cell before computing the fluorescence as we explained before. This is the most common procedure to extract the fluorescence signals [5].

3.1.2 Characterizing

One of the suggestions of the NIG is not to analyze directly the shape of the extracted signals because the high variation of the shapes observed in these signals due to the specific function of the neuron or the underlying spike train which generate that shape. This suggestion has been solved in the bibliography by deconvolving the underlying train of spikes by observing the fluorescent data which these spikes generate.

We will take the same approach of finding the location of the hidden spikes. Although we will work in this direction, we want to be very critical with the results obtained because of some facts:

- NIG won't record simultaneously using fMCI and multi-electrode array, so we won't have any tool to check if the proposed spike's positions are correct or not.
- The type of activity which generate the observed signals make difficult to deconvolve using single spike concept (assumptions explained in the proper chapter, 1.2 Figure 1.7).

- The question of whether precise spike timing information can be extracted from noisy calcium signals is still open debate, especially for fast-spiking inter-neurons. In the previous point we discussed the difficulty of the deconvolution to recover precise spike timing, but there's not only a problem on the difficulty of achieving that goal: there are doubts even on the possibility of making it.

Because of the exposed reasons, our spike detector systems will try to identify as a boolean the spiking activity in a given frame but our main priority is not recovering exactly the original spike train.

3.1.3 Spikes location techniques

Fortunately the calcium activity of the cell may be used to infer the times at which the neuron emits a spike from its cell body due to the fact that immediately after a spike is emitted from the cell, calcium rushes into the cell body and this results in a spike-evoked calcium transient that has roughly exponential decay with a decay time of approximately 0.5-1 second. Due to the long time course of spike-evoked calcium transients and high levels of photon shot noise incurred during the sensing process, the problem of inferring spikes hidden within the calcium activity of a cell is extremely challenging. Furthermore, the question of whether precise spike timing information can be extracted from noisy calcium signals is still open to debate, especially for fast-spiking inter-neurons.

Some different approaches has been used to infer the location of the original spikes. We will discuss some of them in the next points.

3.1.3.1 Statistical Modeling Approach

The paper [19] is a strong pillar in the bibliography related with calcium imaging analysis. The paper describes a model to generate synthetic data with fluorescent characteristics. The paper starts presenting a model with several restrictions and assumptions and continues generalizing the model to make it more flexible and suitable to generate generic signals. Despite the first part of the paper describes a generative model, it continues developing a system to infer the underlying spike train which most probably generate an observed fluorescent signal.

We will briefly explain the main points of the paper because both parts of the model (generation and prediction) have been used in the implementation of the project. Actually, we will use exactly the same algorithm explained in the paper to infer the spike train and a modification of the generation model proposed in the paper to synthesize our dataset.

- Statistical signal modeling

  The model assume the fluorescence data as observations sequence called F. They define this fluorescence as a non-linear function of the calcium activity Ca^{2+} plus an offset and noise. The exponential behavior of F when an spike occur is given by Ca^{2+}, so the layer which connects F and Ca^{2+} is used to:

  - Explain the saturation of the signal when a lot of electrical activity happens. This saturation is modeled using a non-linear function $\left(S(x) = \frac{1}{1+x^d}\right)$.

– Add some additive noise due to optical phenomenon, maintaining the "real" or hidden event Ca^{2+} and the observed one clearly separated.

F_t can be expressed in terms of a hidden variable Ca^{2+}, of an offset value β, the variance of the noise σ_F^2 and a normalized Gaussian random variable $\epsilon_{F,t}$ as expressed in 3.1.

$$F_t = \alpha S\left([Ca^{2+}]_t\right) + \beta + \sigma_F \epsilon_{F,t} \tag{3.1}$$

Ca^{2+} is defined to have an exponential behavior which tends to a fixed minim value $[Ca^{2+}]_b$. This signal is excited each time an spike happens, which is modeled as a binary sequence n_t which represents whether a neuron is spiking or not in t. Ca^{2+} sequence is considered to be contaminated by an additive noise. This behavior can be summarized with the expression 3.2, where τ set the decay constant for the exponential, Δ is the inverse of the frame rate (so $\frac{\Delta}{\tau}$ makes the constant τ frame rate-independent), A is the impact on the calcium amplitude due to a single spike, $\sigma_c^2\Delta$ is the variance of the noise and $\epsilon_{c,t}$ is considered to be a normalized Gaussian random variable.

$$[Ca^{2+}]_t - [Ca^{2+}]_{t-1} = -\frac{\Delta}{\tau}([Ca^{2+}]_{t-1} - [Ca^{2+}]_b) + An_t + \sigma_c\sqrt{\Delta}\epsilon_{c,t} \tag{3.2}$$

In the mentioned paper the spike train n_t is supposed to follow a Bernoulli distribution of parameter $p\Delta$ as shows the equation 3.3.

$$n_t \backsim \mathcal{B}(n_t; p\Delta) \tag{3.3}$$

This model is based on solid biological theory and is cited in almost every paper which works on this kind of signals and specifically in all those papers which deals with spike/event identification. Some of them even use the notation of this paper and the separation of the F and Ca^{2+} domain to model their own signals or justify their algorithms. Because of that we have considered this generation model significant enough to build our synthetic models on it.

- Inferring spike trains by estimating hidden variables.

  This same paper explains some methods to infer the hidden variables n and Ca^{2+}using the observation sequence F. To do it, they explain the relation between these three variables using a Hidden Markov Model (HMM). They model the sequences F_t, n_t and Ca_t^{2+} as dependent on time. The hidden state in time t is described as a tuple (Ca_t^{2+}, n_t), so its distributed continuously over the domain of Ca^{2+} and discretely over the binary random variable n_t. The observation O_t corresponds directly to F_t, so the observation state is a continue random variable distributed over the domain of F.

  The basic assumption of an HMM are:

  – Each observation is independent of the previous one: $P(O_{t_1}|O_{t_2})\,|_{t_1 \neq t_2} = P(O_{t_1})$

  – The hidden state on t only depends on the previous state: $P(H_t|H_{t-1}, H_{t-2}, ..., H_0) = P(H_t|H_{t-1})$

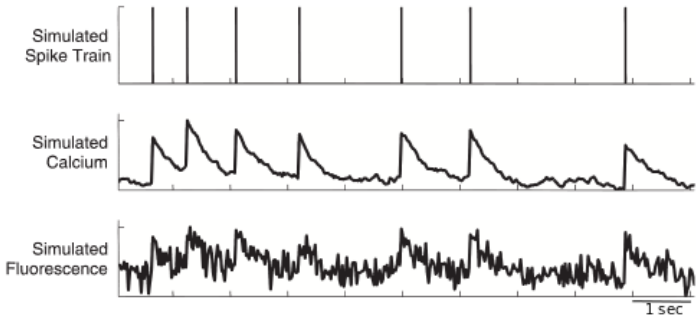


Figure 3.1: This capture taken from [19] shows how the algorithm starts generating a sequence n of spikes (in this case following a Bernoulli distribution), then computes the calcium signal corresponding to the generated train of spikes and finally the observation fluorescence sequence result of a non-linear transformation of the calcium signal plus additive noise.

The model is based on a number of time-varying states, each governed by a set of constant parameters θ. We show distribution required to describe a HMM: joint probability distribution of hidden states and observations (eq 3.4), probability of a given observation given a hidden state (eq 3.5) and probability of a given state given the previous one (eq 3.6).

$$P_\theta(O_{1:T}, H_{t:T}) = P_\theta(H_0)\prod_{t=1}^{T} P_\theta(H_t|H_{t-1})P_\theta(O_t|H_t) \tag{3.4}$$

$$P_\theta(O_t|H_t) = \mathcal{N}(F_t; [Ca^{2+}]_t, 1) \tag{3.5}$$

$$P_\theta(H_t|H_{t-1}) = \begin{cases} \mathcal{N}\left([Ca^{2+}]_t; f\left([Ca^{2+}]_t\right) n_t, \sigma_c^2\Delta\right)(p\Delta) & \text{if } n_t = 1 \\ \mathcal{N}\left([Ca^{2+}]_t; \sigma_c^2\Delta\right)(1-p\Delta) & \text{otherwise} \end{cases} \tag{3.6}$$

The mathematical development has been done using the assumptions explained in the previous point of this project. This mathematical development corresponds to the simplest scheme proposed by the paper, a scheme using just a linear relation between $Ca^{2+}$and F ($S(x) = x$) in the expression 3.1). Although this is the simplest model, we consider it's enough to understand how this algorithm works.

Once these distributions has been defined, the paper try to compute the value of $P_\theta(H_t|O_{1:T})$. This is the real important distribution we want to compute because if we can find the sequence of hidden states which most probably generate the observations F, we will find the sequence of spikes n_t which most probably generates F, which is precisely the objective of a spike detection technique $\left(\underset{n_t}{argmax} \quad P_\theta(O_{1:T}|n_T)\right)$.

The problem of estimating this distribution is that it cannot be computed analytically with the available information. The technique used in this paper is 'particle filter', which consist on sample the distribution H_t to get the value of that distribution on a subset of specific values. The values which must be sampled are selected proportionally to the estimated Probability Density Function (pdf) or sampling function of that hidden state [20, 3, 4]. This Monte Carlo methods or

particle filters are usually used when the state space of the latent variables is continuous rather than discrete and the distributions of the hidden variables are not Gaussian (in which case, we could use a Kalman Filter).

We actually spent some time trying to implement this algorithm, but the implementation is quite complex specially when the model is made more flexible and it didn't work at all. Finally we decided to contact with the authors of the paper who kindly sent us their implementation, which we have used for this project.

Since we finally just used the implementation the authors sent us, we won't give more details about it.

This statistical approach is very powerful and allows us to infer spikes from models which fits reasonably the proposed model. One of the strong points of the algorithm is its capability of learning the parameters to really fit the signal we are working with.

The problem of this algorithm is its strong dependence on the model generation assumptions. In the cited paper they assure that the results are good enough when some assumptions are relaxed and that the model can be used to find spike trains with a minimum setting of the initial parameters.

The problem we have found is that our model assumes bursts of spikes in short periods of time, which is an assumption very different of a spike pattern which follows a Bernoulli distribution. Even relaxing some assumptions of the model, we felt that the results when the algorithm was used with real data were not good at all. The alternatives where:

1. Discard the model.
2. Modify the algorithm and the sampling method for the particle, build a new model for our data.
3. Try to fine tune the parameters to maximize the performance of the algorithm.

Since the algorithm described in the paper is quite complicated and the time to develop this project was very limited, modifying the whole sampling algorithm was considered out of the scope of the project. Trying to fine tune the parameters was very difficult because the available signals were very short due to the small size of the dataset, so we could not be sure if we were generating a good model of we were just over-fitting the parameters to the short available data.

We finally decided just to move on and work with some other alternatives to detect the underlying spikes which generates the observed fluorescence process. Nevertheless, we have used the generation model proposed in the paper as a basis for our synthetic data generator algorithm.

3.1.3.2 Unsupervised adapted filter approach

This approach is very simple and is proposed in [15]. The technique consist on, once the fluorescence data has been extracted from a given neuron, use a band-pass filter and choose the K highest peaks in the trace. A template is made by averaging the samples surrounding these K highest peaks on the original data. This template is used as a matched filter by shifting it point by point along the trace and taking a dot product with the signal's change above it's minimum value within the template duration.

This technique is very simple, but its simplicity imply several problems.

It assumes that all the 'events' will have a similar shape. Since the constant decay depends on the calcium behavior when entering and leaving the nerve cells, it sound reasonably. The real problem is the fact of choosing the K highest points in the trace. There's no problem in choosing these points as reference to build the template, because the signals are not expected to be very noisy and the peaks will correspond with a high probability the result of a spike; the problem appears assuming that there will be at least K events in the signals to build the template. If that does not happen, the resulting template won't be useful.

The second problem appears if we consider our specific dataset. This model is tested with signals which follows a model similar to the explained in the previous point [19], sparse in the spike activation and with single spikes (not trains of them). The difference with our signals is that we expect to deal with burst of spikes happening in short time periods, maybe with only one frame observing the phenomenon. In this cases the resulting shape is the result of adding several exponentials with a delay dependent on the pattern of the burst of spikes. If the pattern of the underlying burst of spikes is different (we should not assume it, since we want to build a model as general as possible) in each burst, probably the shape will be different and the built template cannot be representative at all.

This problem will be even more significant when the shape observed in fluorescent data corresponds to a burst of spikes. The problem is when a burst of spikes appears before the shape (with an exponential decay) of a previous burst of spikes has disappeared. In this case, the shape will not be representative and the template will probably not be useful.

As we have discussed, this is a very simple approach which can be used with models similar to the generative model used in the previous point, [19], where sparsity of the spikes and single spike event assumptions are made. Although these signals can be appropriate and are used in [15] with anesthetized mice, they don't seem very appropriate to work with our signals and our final goal because our signals will potentially present some characteristics this algorithm can't deal with:

1. Sparse activation assumption but with a possible eventual overlapping between shapes derived of different bursts of spikes resulting on a mismatch with the built template or making a non-representative dictionary.

2. Burst of spikes generating the observed exponential-like shapes in the fluorescent traces. The shapes in the same signal can present a high variation.

3. The final aim is to use the detection of these underlying events to clusterize the original signals and find some functional structures. We are actually very interested in working with non-active neurons. This algorithm does not work well with signals where no neuron activity happens.

3.1.3.3 Shape adapted approach

This is the last family of algorithms we will discuss. It's based on locating the exponential shapes which occurs in the fluorescence signals when an spike occurs. This technique requires some previous information regarding to the shapes we are expecting to find.

In [10] a decomposition of the original fluorescence signal x is proposed using a dictionary ϕ containing several atoms φ_m consisting on spike-evoked waveform placed at the m^{th}sample and a vector of coefficients α which weights each atom in the dictionary ϕ as shown in the equation 3.7.

$$y = \sum_{m \in A} a_m \varphi_m + \eta = \phi a + \eta \tag{3.7}$$

The paper discuss what are the conditions for exact Spike Recovery and propose several restrictions to improve the efficiency and the quality of the algorithms such as imposing non-negative restrictions on a or the sparsity on a by minimizing the amount of zeros in this weight array when finding the optimal dictionary or imposing some conditions on the sparsity of the hidden spikes.

Since our final objective is working with real data and we won't have any way to objectively measure the quality of the spike detector (it's important to remember that even biologists are not sure of the possibility of recovery these events in real data), the considerations made in the paper are nor relevant for us. We will focus our attention in the algorithm to find where the events happen once the dictionary has been inferred.

The main idea of this algorithm is very similar to the idea proposed in [5], consisting on analyzing the fluorescence signal sequentially and trying to find, in each one of these positions, the best combination of exponentials (inferred when the dictionary is learned in the case of [10] or just trying exhaustively all the exponentials which can be generated modifying some parameters as in [5]). Some criteria is used then to accept or reject this combination of exponentials as the responsible of an event in the analyzed signal. If it's accepted, the estimated exponential shape is subtracted from the analyzed signal and the algorithm continues. In [5] two criterion are used:

1. They define a threshold and they check when the signal reaches this threshold and remains over it for a given time

2. The integral of the remaining signal (once the inferred wave has been subtracted) is close to 0, since a positive defined event has been removed.

In Figure 3.2 a plot taken from [5] explains the iterative idea of this algorithm.

The results achieved with this algorithm are quite good taking into account that can be used even if no spicular activity happen in a fluorescence sequence, it's quite robust to overlapping events generated from close-located spiking bursts and is not really dependent on the shape of the signals, specially if we give enough freedom to choose a wide range of decay parameters for the exponentials in the dictionary.

An idea based on this algorithm has been implemented and tested. The implementation is very simple and don't take into account the optimizations proposed in [10] because these optimizations are made assuming the most simple model presented in [19] as can be appreciated in the Figure 3.3 where ideal sparsity is assumed and there's no saturation modifying the shape of the exponentials shown.

As our goal is to check the potential of the the proposed complete method to localize functional structures by analyzing fMCI images and the detection of spikes is just one of the parts, we consider that a very rudimentary algorithm is enough to test this potential. Once longer data will be gathered and available to be analyzed, the election of the spike detector algorithm and implementation should be reconsidered.

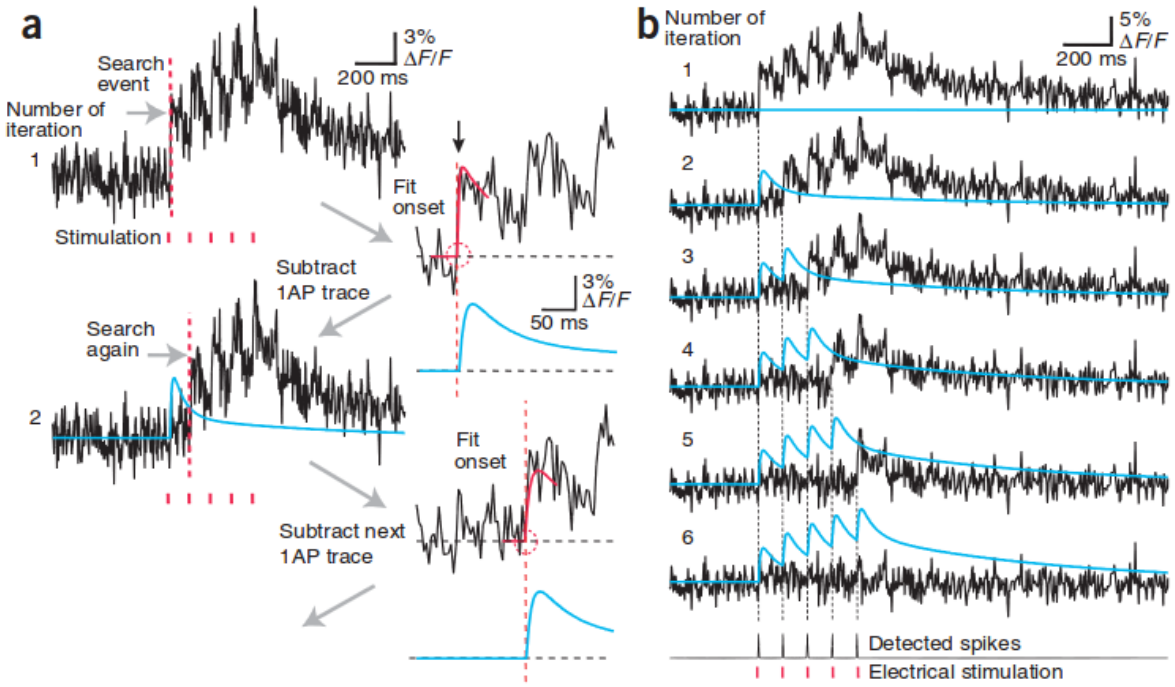


Figure 3.2: Plot and legend taken from [5]. (a) Illustration of the automated peeling procedure. In the initial step, a first 'event' is detected using a customized Schmitt-trigger threshold routine. The onset of the detected calcium transient is fit within a short time window (red curve) in order to obtain a good estimate of the starting point of the event (red circle). Then a stereotypical 1AP-evoked calcium transient is placed with its start at this time point and is subtracted from the original fluorescence trace. Then the algorithm starts the next iteration on the resulting difference trace. (b) Example of the extraction of a train of five action potentials evoked by electrical stimulation at 10 Hz. For each iteration, the residual fluorescence trace as well as the accumulated trace of all 1AP-evoked calcium transients thus far extracted (blue trace) are depicted. After five iterations, no additional event was found in the residual fluorescence trace. At the bottom, detected spike times are shown together with the stimulus times.

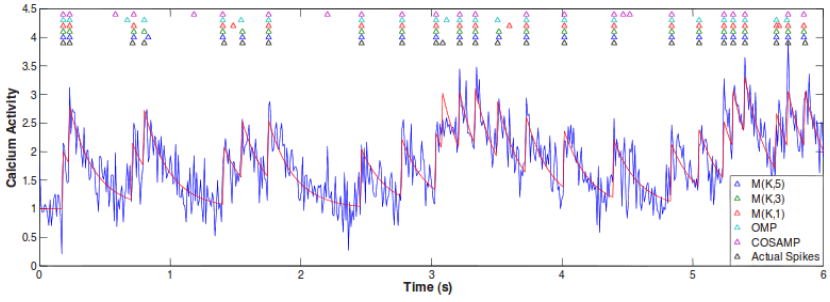


Figure 3.3: Results achieved using the exponential fitting algorithm explained in [10]. As we can see in the image, the algorithm seems to work very well, but the data they are analyzing looks not as real data: there's no saturation and the events looks like perfect exponentials generated by a single spike, not an addition of several exponentials generated by bursts of spikes.

3.2 Finding functional clusters

In the previous points of this chapter we have made a review of the bibliography to extract the fMCI temporal signals corresponding to individual neurons from the fMCI videos and some techniques to infer the underlying spiking events which generates the observed fMCI series. In this point the final aim of the project is discussed: how to find the relation between the activity of the different neurons analyzed.

The desirable properties for the technique we want to work with should be:

- It should be independent on the shape-variation due to the saturation of the signals caused by the shape of the original located neurons.
- It should be unsupervised.
- We should be able to generate some synthetic model based on reasonable assumptions to test our algorithms before start trying it with real data.

There's not much bibliography related with this topic; probably because the difficulty related with the validation of the proposals. As we have explained in previous points of this report (1.1) the algorithms to infer where the spikes happens can be validated by analyzing simultaneously the neuron activity with some electrodes and with fMCI techniques, validate if some neurons are or not connected or related is more complicated. One way to do it is physically test it, which implies the dissection of the studied animal. There's some restrictions on this way to experimentally verify the obtained results:

1. Changes in the physiology of the animal due to its age. The animal must be dissected just after recording the video to be analyzed; but the video must be processed and analyzed to know beforehand what must be checked when dissecting the animal.
2. Two neurons can not be physically connected but some relation in their activity can exist. In this point we are dealing with a problem similar to determine the difference between causality and correlation. Two neurons can be related directly or may exist some hidden stimulus which stimulates both neurons.

We have not found any paper where a quantitative metric is given to validate the algorithms to clusterize the analyzed signals to find groups of neurons which are related when they are used with real data.

Instead of that, they generate synthetic data relating the different neurons and then they try to recover this relation using several techniques. Often, they discuss how the proposed algorithms work with real data using as arguments biological criteria such as the function that biologist consider that some cerebellar microzones are responsible of. Anyway, it's important to say that not only the validation method is difficult for technical reasons; it's difficult because there's not a known 'map' of the functions of the brain and neurons: this is the reason of this research, help biologists providing them some tools to automatize the analysis of populations of hundreds of neurons in order to improve the knowledge they already have.

In [11] they explain these issues more accurately and propose the possibility of finding some relation between neurons which correspond with stable anatomical entities.

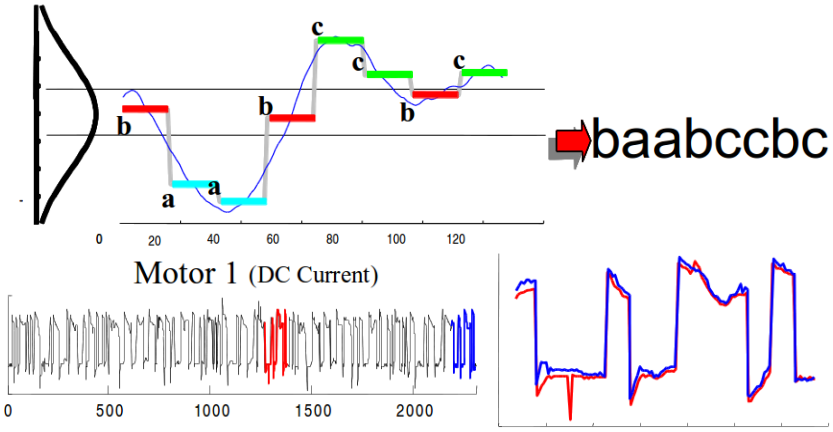


Figure 3.4: Images taken from the friendly ppt explanation of SAX (http://www.cs.gmu.edu/~jessica/sax.htm). In the top image a signal is transformed using the proper dictionary. In the bottom image we can see an example of a motif extraction in a temporal sequence.

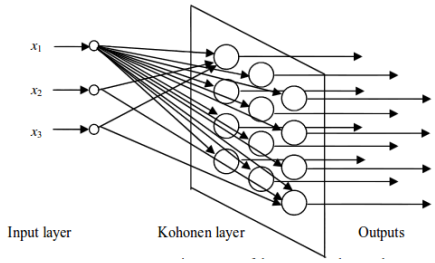
Other students in previous internships in the NII have faced the problem of finding relations between the activity of different neurons by finding some common patterns in the analyzed data. These techniques are widely used in financial time series analysis to find patterns which often appears in a sequence; these patterns are called motifs. Techniques as Symbolic Aggregate approXimation (SAX) [18], based on summarizing temporal series by quantifying these signals with dictionaries of low-symbols dictionaries; variations of Pulse Amplitude Modulations[PAM]), are used to find these motifs (Figure 3.4).

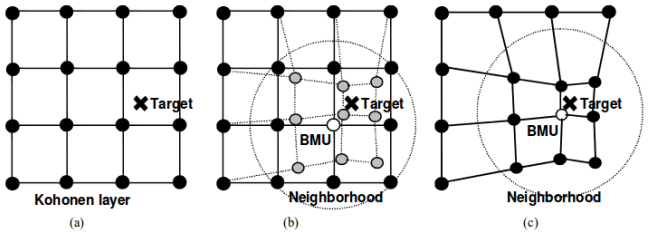
Although there's a lot of bibliography which explains the limitations of this techniques and which propose some solutions to avoid these limitations ([2]) this family of methods did not reach satisfactory results to these previous students.

When we have experimented with these techniques, we have found that the patterns which appear in the time series extracted from fMCI are not very "regular" or "stable" and the variation in these shapes is very significant. This may be explained by:

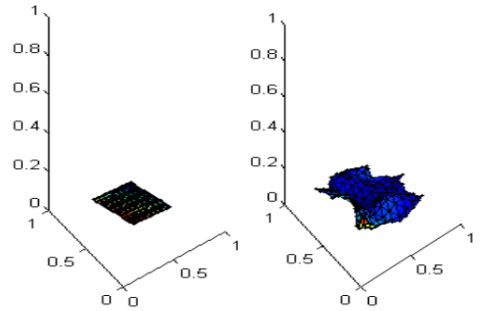
- The low frequency rate we are working with: the events are very short and summarize them using a dictionary with a few symbols implies that we can sample very different amplitudes depending on when we started sampling.
- This technique require long signals and a lot of occurrences of a given pattern to distinguish it.
- For clustering techniques, it would be interesting that the shapes observed in different neurons would be very similar. As we explained in the point 1.3 our real data does not allow us to make this assumption.

We explored Self Organization Models (SOM) techniques, which can be used to represent high-dimensional data in 2D space but which is used in [16, 36, 9] to clusterize climate datasets where exist a dependence not only in the observed data but in the location of the spots where the data is gathered.





(a) Basic structure of the SOM neural network. (b) Illustration of the SOM network learning: (a) Initial state, (b) state at step k, (c) state at step k+1



(c) Illustration of the SOM neural network results given by [16]. Initial random distributed SOM states (left) and SOM outputs after training (right) using 29 geographical x-y positions and their corresponding meteorological data (temperature and precipitations).

Figure 3.5: Some plots taken from [16] which helps to explain how SOM algorithm can be used to cluster temporal series.

The main idea of this algorithm consists on building a Neural Network (NN) with 2 layers. The input layer corresponds to M neurons being M the features of the samples we want to clusterize and the second layer corresponds to K neurons where each neuron j has associated a "synaptic weight vector" w_j in the domain of the input layer (Figure 3.5a); so each neuron in the second layer is fully connected with the input layer. The particularity of this scheme if that not only the weight is important, but the position of each neuron. The training process of these NN is, for each sample and for as much epochs as desired (Figure 3.5b):

1. For each neuron j, compute a defined distance criteria with the input sample x and find the Best Match Unit (BMU) which is the weight or reference vector w_{BMU} such that:

$$w_{BMU} = \underset{j}{argmin} \|x - w_j\|$$

2. The weighted vectors are updated. Here is where the location of each neuron is used and makes this technique different of the classical NN: the weight of each neuron is modified depending on the distance to the input vector and the distance to the w_{BMU}.

$$w_j[k+1] = w_j[k] + \alpha[k] h_{bj}[k] \, [x - w_j[k]]$$

Being $\alpha[k]$ the learning rate and $h_{bj}[k]$ the neighborhood kernel centered on the winner unit w_{BMU}.

We thought that in our system something similar happened because we have a similar starting point and similar objectives. The main problem of this algorithm is that, despite the output is nice and can be interesting for some applications like the one showed in the referenced paper, it does not seem to give good results if the starting mesh is not dense enough. Another problem with this algorithm is that a implicit condition is imposed between physically close neurons.

This last fact is what made us to discard this family of algorithms. We are very interested in relating the electrical activity of the neurons independently of their location in the analyzed images; we must take into account that we are only observing an slice of the tissues of the analyzed animal. This means that two close neurons can be absolutely unrelated if they are not connected and two geographically far neurons can be connected in a non-observed slice so the location of the neurons in the observed images is not so important and should not have a direct impact in the chosen algorithm.

After some analysis with these techniques we concluded that it can be a useful technique to make nice visual representations of this clusters (Figure 3.4) but is not useful at all for our purpose because it set some implicit assumptions we want to avoid.

We decided to find specific alternatives for the signals we are working with. The technique we have taken as a reference to start working is based on [15].

There are some important points in this paper which are typical in most of the related bibliography and which allow us to reach the objectives we set for the clustering algorithms at the beginning of this point:

1. Neuron connection model

   The neuron connection model they use is a very simple model based just in a matrix M of size NxN. Given N the number of neurons which are being analyzed and F_i the fluorescence calcium signal corresponding to the $i-th$ signal, the position $[i,j]$ of the matrix correspond to $\rho_{i,j} \in \mathbb{R}^1$. In this model, the probability of a spike happening in F_i causes a spike in F_j is given by a Bernoulli distribution of parameter $\rho_{i,j}$.
   As we can see, this model is based on an strictly Bayesian approach which does not consider the possibility of any given events. The model consider that an event can appear spontaneously or be caused by another event which has spontaneously appeared in another fluorescence signal.

   This can be expressed in the following way:

$$n_i = n_i^{'} + \sum_{i \neq j}^{N} B\left(\rho_{i,j}\right) n_j^{'}$$

   Where $n_i^{'}$ represents the spontaneous speaking of the $i-th$ neuron and n_i the spike train which generates the fluorescence signal F_i with a given non-linear function G ($F_i = G(n_i)$). A very similar approach is used in [38] and because of these two papers are the most directly related with our final objective, this connectivity model will be used as a base for the model we will use in this research.

2. Spike detection as preprocessing step

   In the studied literature, they insist in the need of start finding what the spike activity of the neurons happen before start clustering the data. Despite it's not argumented at all why this step is necessary, we guess is because of the high variability in the fluorescence images extracted from fMCI. One of the reasons of the variability in the shapes which appear in the extracted signals is due to different spikes which originate the observed sequences, but sometimes it's only because the shape of the neuron is different, the delimiting zone of the neuron is not well detected or the studied neuron has an offset saturation of calcium. It's important to say that the fact that the neuron is saturated with calcium does not only imply an offset or baseline in the observed signals; if we understand the relation between the spike series and the fluorescence signals as described in [19] this relation is non linear and exist some saturation in the fluorescence imaging which can be the cause of different shapes in the final fluorescence signals.

   We will discuss this in the results analysis of this project, but since this step seems so important in the related bibliography we will work directly with the original signals and with a preprocessing consisting in a spike-deconvolution algorithm as explained in 3.1.2.

3. Clustering process

   There are some different techniques in the bibliography to infer the underlying relation between the different neuron activity signals. In [38] is discussed how to find this relation extending the algorithm explained in [19] which explains how to understand the generation process as a HMM and the infer process as a parameter estimation.

   In the early spike detection tests this algorithm does not seems to be very flexible to data which is not generated exactly by the proposed model. To define a new statistical model and how to infer the underlying or hidden parameter using Monte Carlo techniques are out of the scope of this project so we won't work with this approach.

   The approach we will work with is a simpler approach which is based on a very simple k-means algorithm. The signals used for this k-means algorithm are just a train of spikes resulting of analyzing the fluorescence signals. The metric used to compute the distance between n_i and n_j, where n_i is the inferred spike sequence corresponding to the $i-th$ fluorescence signal F_i is the Pearson correlation:

$$r_{i,j} = \frac{\langle n_i, n_j \rangle - \sum_{z=1}^{N} n_i[z] \sum_{z=1}^{N} n_j[z]}{\sqrt{\langle n_i, n_i \rangle - \left(\sum_{z=1}^{N} n_i[z] \right)^2} \sqrt{\langle n_j, n_j \rangle - \left(\sum_{z=1}^{N} n_j[z] \right)^2}}$$

   In this paper they propose not to use directly n_i for compute this distance metric because the low frequency rate and the high firing rate of the neurons makes the position detected non meaningful at all; a difference of one or two samples can make a big difference when $r_{i,j}$ is computed but it does not represent the real distance between these two signals.

   To solve this problem, the paper propose a likelihood-based correction system for event times which consist on substituting an event detected on time t_z represented by an amplitude A by two events (Figure 3.6) :

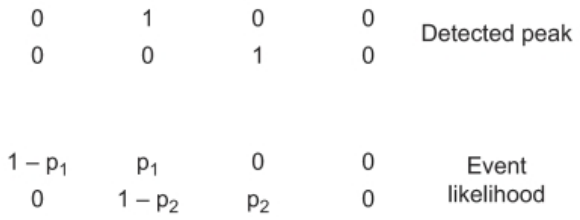


Figure 3.6: Example of likelihood-based correction system proposed by [15].

Algorithm 3.1 Basic k-means algorithm

```
1 x <- signals to be clusterized
2 r <- k centroids which initializes the algorithm
3 C <- clusterss
4 while cluster change respect previous iteration:
5     for sig in x:
6         add(sig,nearest centroid in C)
7     recompute r
```

- An event which happens in t_z with an amplitude $A(1 - x/L)$.
- An event which happens in t_{z-1} with an amplitude $A(x/L)$.

Where $0 \leq x \leq L$.

Once the metric and the preprocessing process over the fluorescence signals are defined, a k-means based algorithm is run.

The proposed scheme is based on a $k-$means algorithm; all the samples are clustered in k clusters (k is defined before starting clustering). A $k-$means algorithm understand the input signals $n_i = (n_{i1,}...n_{iL})$ as a point in a $\mathbb{R}^L$ space and is initialized with k samples in $\mathbb{R}^L$. The algorithm runs as a simple EM algorithm described in 3.1.

The paper describes an algorithm which is a 'meta k-means' algorithm. It consist on repeating the k-means algorithm M times with different initializations and after that, compute for each pair of signals n_i, n_j how many times they have been clustered in the same cluster. They recommend to use a value M of 1000 iterations and a value k equal to 3.

In the Figure 3.7 we show one of the plots in [15] which summarizes the described process.

When we experimented with this method, we felt that the fact of setting a fixed value for k imposes big clusters when hundreds of neurons are being analyzed. The probability of distinguish small clusters with not many neurons in each one or even find not very strong relations is very low and the algorithm didn't got good results in those cases. In the following points of the project some new ideas are discussed and tested to solve this problem.

This proposed approach is quite general and don't assume neither any connectivity restrictions dependent on the location of each neuron nor other previous information. Anyway there's a big assumption which is done here, and is the assumption of some independent events which are spontaneous and

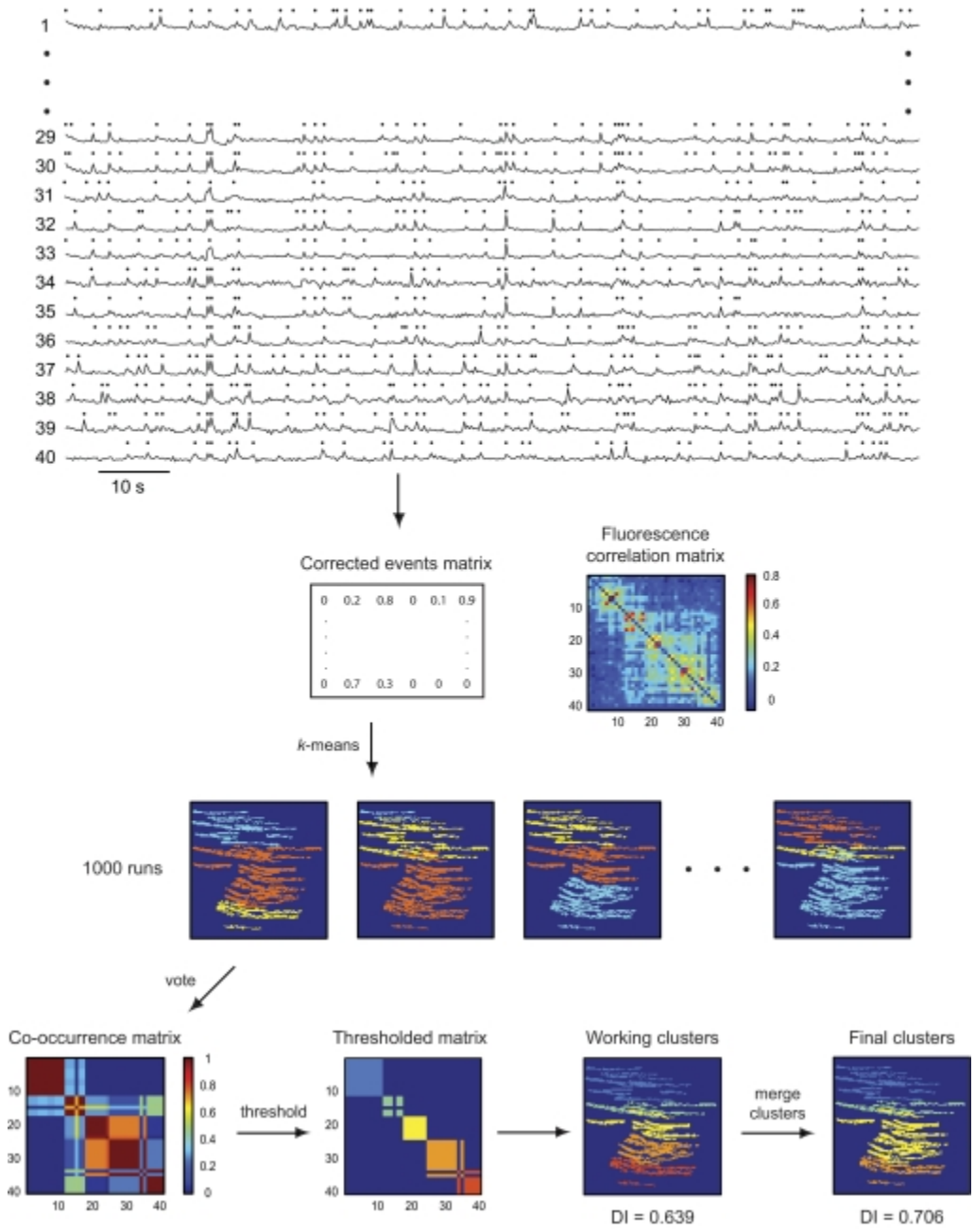


Figure 3.7: Figure extracted from [15] which shows how the proposed meta-k-means works. A selection of traces is shown in the top row. The dots indicate detected events. K-means is run 1000 times on the corrected events matrix using k = 3 and random initialization of centroid positions, yielding several different clustering results (3rd row). The co-occurrence matrix (bottom row), determined by the number of times each pair of dendrites is clustered together is similar to the fluorescence correlation matrix (2nd row), and is thresholded to give working clusters (bottom row).

some events which are inducted by the influence of the spontaneous events produced in other neurons; using this model and representing it by a matrix filled with something very close to a weight for each connection, the relation between neurons can be understood intuitively as something similar to a cross-correlation matrix.

Under this assumption it seems clear that k-means using pearson correlation will be able to clusterize in the same cluster signals which are 'close' using that metric and using an appropriate threshold the result of this meta-k-means algorithm will be an approximation of the matrix which we have used to generate the data.

Although biologists are not sure about the underlying structure of neuron connection yet, we consider that the model proposed in these papers is interesting because it allows us to work in an absolutely unsupervised way, can be used starting only from the fluorescent data, is quite general, and allows us to check the good or bad behavior of the proposed algorithms by generating synthetic data.

Independently of the final results, the question here is: is this model representative of the real events? The activity between all these neurons is propagated under this Bernoulli assumption?

Chapter 4

Proposals and implementation

In this chapter we will detail how a system to reach the objectives explained in the point 1.4 has been structured and built. The techniques implemented are the techniques which have been explained in the chapters2 and 3 or modifications made using them as a base.

We will focus the explanations in this chapter to the modifications made in the basic algorithms discussed in 2 and 3. We will justify which are the benefits of the proposed modifications in front of the original ones but we won't repeat in each case the discussion of the limitations and strong points of each alternative.

4.1 General Scheme

In Figure 4.1 we resume the whole process and the steps which are required to reach our final objective. Following this general scheme as represented in Figure 4.1 the systems we have to implement takes as input one of the items labeled with a number and as output the next item.

The main steps in this process are:

- A) Record the zone of interest of a zebrafish with fMCI techniques.
- B) Correct the movement of the fish while it was being recorded.
- C) Locating automatically the neurons which appears in the recorded videos.
- D) Extracting the signals corresponding to the activity of the detected nerve cells.
- E) Finding the relation between the detected neurons by analyzing their activity.

The work required in step A corresponds to the NIG biologists, so there's non implementation required on it. We must provide an implementation for the rest of the steps, which we want to automatize. Since A is an step which does not depend on us, we assume that each one of our samples will consist on a set of images corresponding to the frames which conform the videos recorded by the NIG.

One of the main limitations we have faced in this project has been the lack of enough material to test and even to statistically model our systems. We have implemented our algorithms by analyzing

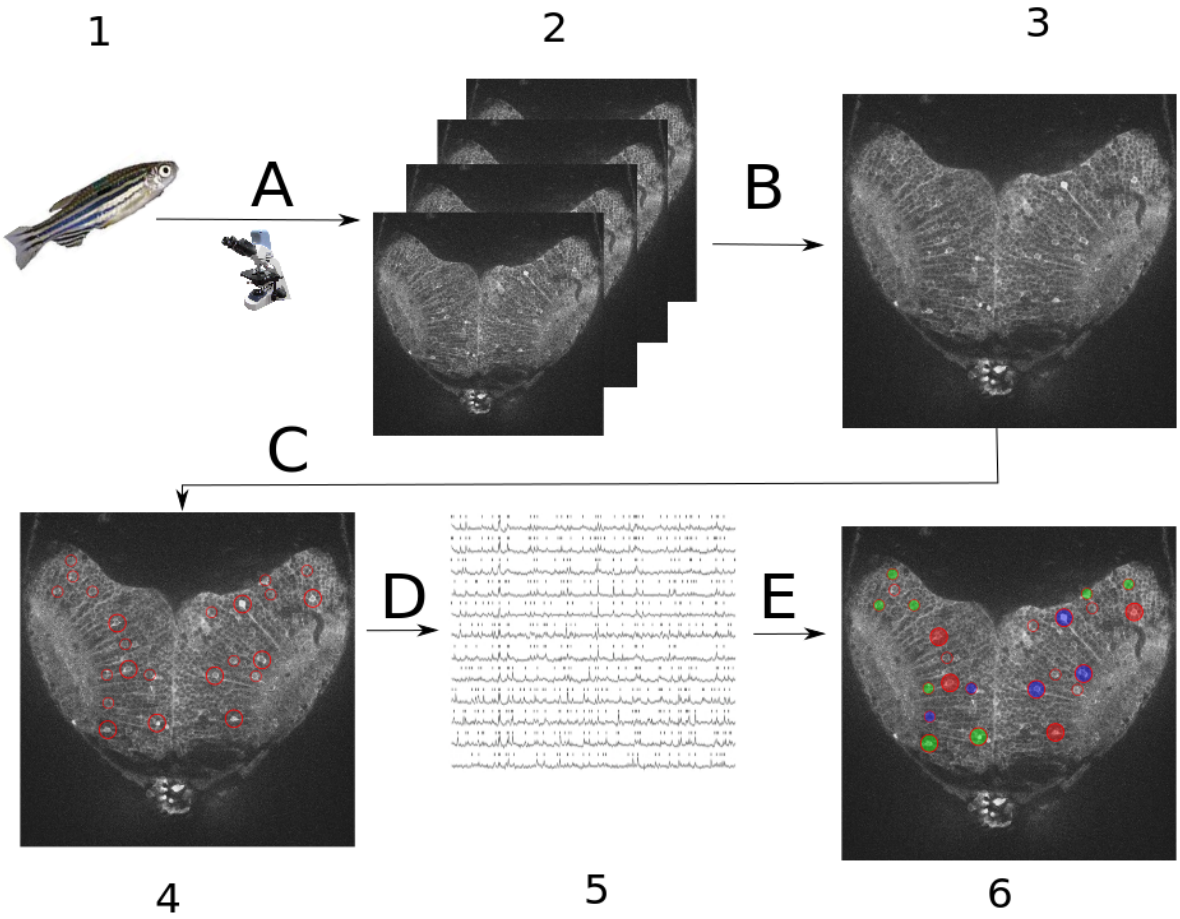


Figure 4.1: General scheme describing the steps our system must support. Our system will be implemented as a pipeline where the algorithms and methods are represented by arrows and its input and outputs by a number. 1) Represents the original input, the fish to be analyzed. 2) is the set of images extracted from the video which results of applying the fMCI technique. 3) represents a set of images where the possible movement of the fish has been removed, i.e., a neuron is located in the same coordinates in all the frames of the corrected sequence. 4) represents the neurons detected in the video. 5) Represents the fluorescence signals extracted from each identified neuron. 6) The neurons are classified in functional groups; in this representation each group is represented by a color and the possibility of some neurons non-clustered is considered.

carefully the bibliography and proposing systems as generals and flexible as possible in order to make the future work of adapting these algorithms to the specific characteristics of the signals once they are be available as easy and fast as possible. Although we have taken into account the NIG recommendations and the bibliography we have found, we must define some framework to objectively evaluate our algorithms and if they are or not promising.

We considered several alternatives such as using datasets recorded in mice, but NIG biologists didn't like this idea because they considered that the neural activity in mice and visual cortex of zebrafish is very different so we finally rejected this option. The framework we defined consists on generating synthetic data which try to represent as better as possible the real data we currently have. This framework is very useful because:

- Allow us to work with longer data. The current dataset consist only in 1 video of 120 frames (12 seconds). The expected length of the videos recorded by the NIG with their new microscope is at least 1 minute (600 frames).
- Allow us to have information of the original stimulus which generate the analyzed data. This allow us to implement some metrics to check is we are whether or not able to recover correctly this hidden information.

4.2 Pipeline structure

In Figure 4.2 we show the general structure of the system developed. It has been implemented as a pipeline with the inputs and outputs represented in Figure 4.1 (from step B to E). This scheme allows us to check the results and plot these results.

Actually the testing and plotting blocks are not considered basic for the system since they have only sense in some configurations; the testing functions only has sense if the Signal Loader is set to work with synthetic data and the plot signal only if the Signal Loader works with real data (at least with our systems because our synthetic signal generator only generate traces, not sets of images).

This pipeline structure allows us to implement several alternatives for each one of the 'boxes'. We have considered that is a good practice to design a system following these schemes because it allows us to easily check the impact of a specific part of the system in the performance of the whole system.

In the following points we will describe the implementation proposed for the different blocks in the pipeline. The described blocks used properly will allow us to configure the system with the desired algorithms and configure it to work with real or synthetic data.

4.2.1 Signal Loader

The general scheme shown in Figure 4.3 represents the outputs of the Signal Loader module. The only restriction for a Signal Loader implementation is its output: a set of N numeric arrays. The other outputs can be implemented if the testing modules or the result plotter modules are used, but they are not mandatory.

Two families of Signal Loaders has been implemented as shown in Figure 4.3: one which uses synthetic data and the other which extract the signals from fMCI sequences.

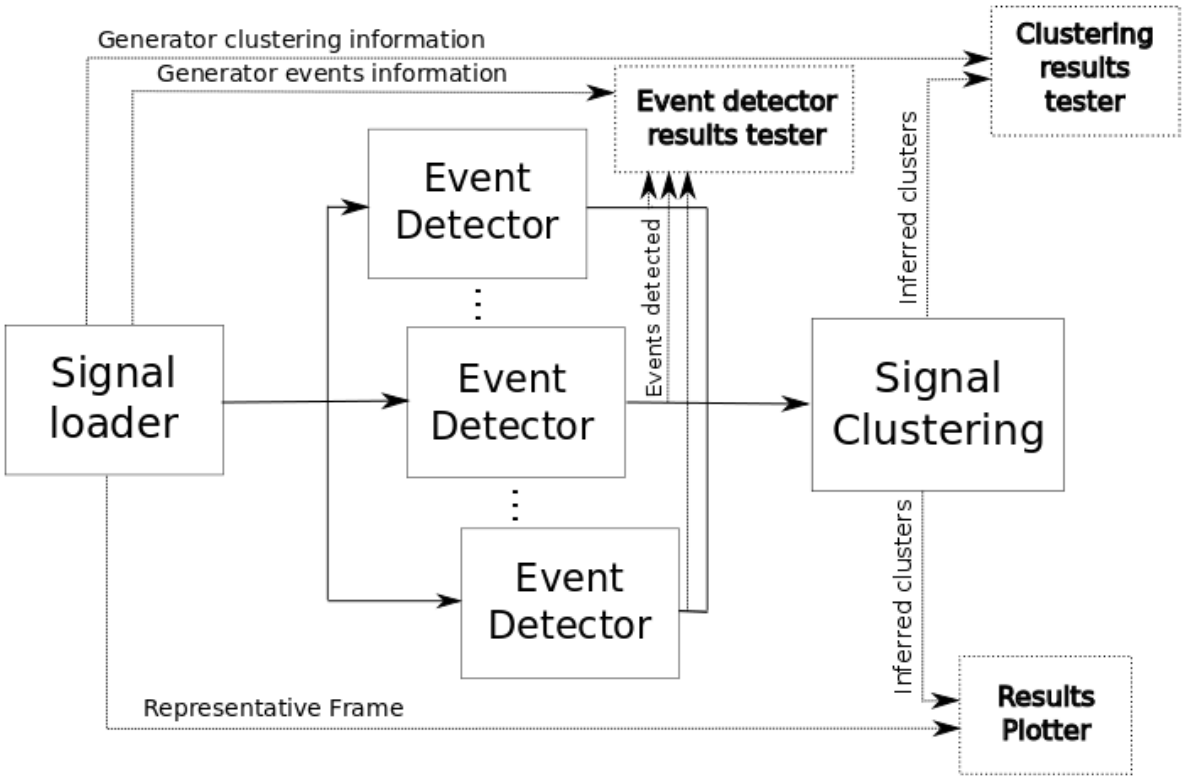


Figure 4.2: General block structure of the system. The solid box and arrows describe the basic part of the algorithm and the dotted boxes and arrows describe the parts of the system which not always can be used.

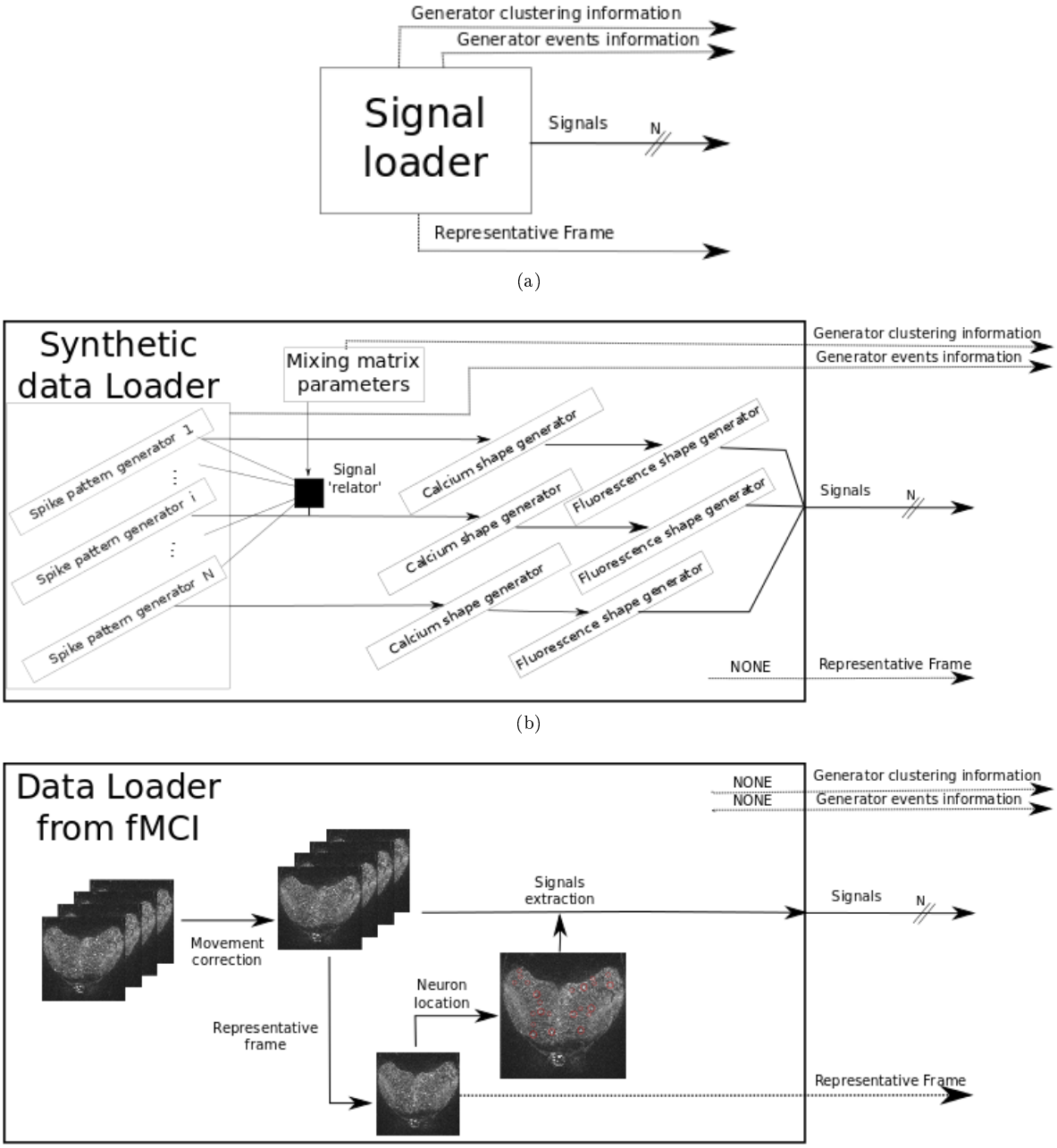


Figure 4.3: Design of Signal Loader modules. a) Design of a Generic Signal Loader. b) Implementation of Synthetic Signal Loader based on the proposed Generic Signal Loader. c) Implementation of Real Signal Loader based on the proposed Generic Signal Loader.

4.2.1.1 Data loader from fMCI

The process is explained in the Figure 4.3 c). First of all we correct the possible movement of the fish during the recording of the fMCI video. In our implementation, we are not using any algorithm to correct the effect of the fish movement because the signal we have does not present important movements and a given nerve cell is almost in the same coordinates in the whole movie.

A representative frame is extracted from the motion corrected frames. We compute this representative mean just by computing the mean of all the video's frames. We find then the location of the neurons and, once we have a set of areas A where a_i corresponds to the the set of coordinates which are part of the $i-th$ detected neuron.

Since we assume the possible motion in the movie has been corrected, given a set of frames X, the output signal associated to the $i-th$ detected nerve cell y_i is computed as:

$$y_i[j] = \frac{\sum\limits_{\forall s \in a_i} X[j][s]}{\sum\limits_{\forall s \in a_i} 1} \tag{4.1}$$

The only point in the scheme which is not explained yet is how we detect the neurons in this representative image. As we introduced in the chapter 2 we use a combination of the watershed algorithm and a dictionary based model. This is not the major of the project and the algorithm is still under study by other Intern Student in the NII. Anyway, we will give a general explanation on the general idea.

The approach makes the following assumptions:

- In the video there are, at least, some neurons which show some activity in the analyzed video.
- The shape of the neurons appearing in the recorded video are similar.

The first assumption is used to unsupervisedly find the active nerve cells in the video. This is done by computing the variation in the intensity on each frame and finding the regions with strong variance in the bright of their pixels. We will consider the selected zones as blinking neurons.

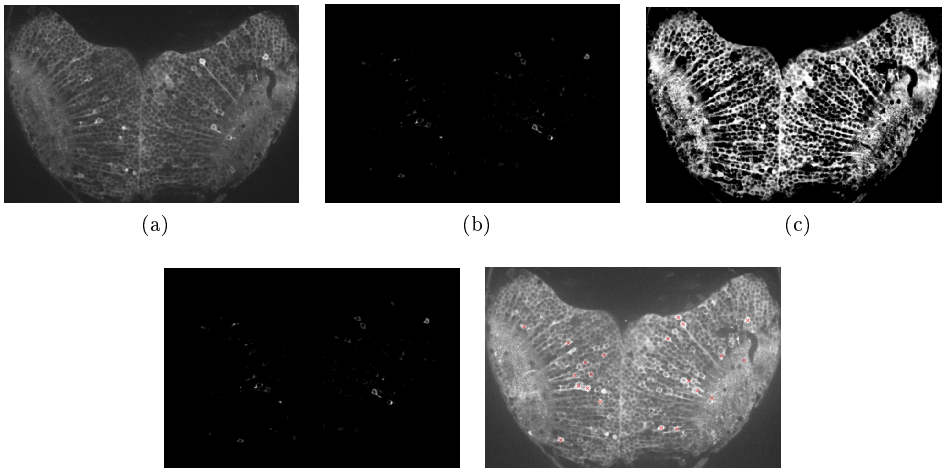
The output is a set of coordinates BN where bn_i corresponds to the centroid of the $i-th$ blinking neuron. We use this to build a dictionary φ. The method to build each word φ_j is nowadays under experimentation and not an specific technique has been definitively chosen (personally I think it would be interesting to try some deep learning technique such as auto-encoding).

Once the dictionary φ has been built, a set of candidates is compared with this dictionary and a decision is taken: a given candidate is or not a neuron.

These candidates are found with watershed segmentation, which allows us not only to know the position of each candidate, but the area corresponding to each one of them in order to use the expression in 4.1.

The prototype provided by the person who is working on this part uses a simple euclidean distance and a feature to build the dictionary is just using the pixels of a zone surrounding each neuron. This method is very primitive, so the located neurons must be checked manually in order to avoid mistakes.

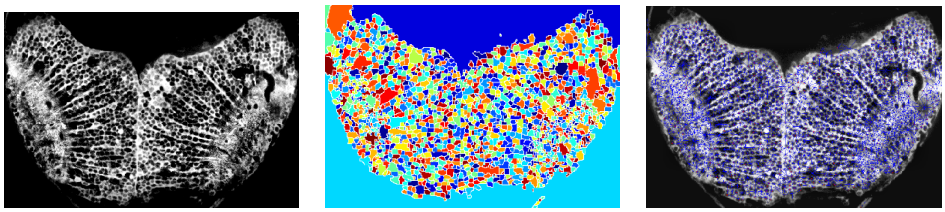
In future some more specific algorithms should be explored specially to feature extraction and to avoid rotation or scale dependence on the shapes which are searched on the images. Some plots which



(a) (b) (c)

(d) (e)

Figure 4.4: a) Original frame. b) Sequence background. c) Background extraction plus contract enhancement. d) Blinking zones. e) Extracted neurons



(a) (b) (c)

Figure 4.5: c) Background extraction plus contract enhancement. d) Watershed segmentation; each color represent a segment found using watershed segmentation technique. d) Candidates to check with the dictionary of extracted neurons.

illustrate the main steps of the process are shown in Figures 4.4 and 4.5.

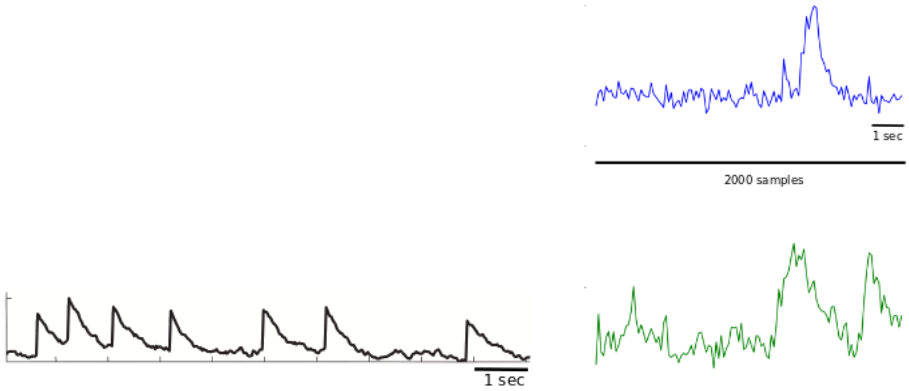
4.2.1.2 Synthetic data loader

The synthetic data generation has been based on the paper [19], explained in previous chapters. We have implemented the algorithm following the same indications of the cited paper and we have proposed a variation of this generation algorithm to adapt our signal to the expected firing rate of the analyzed zebrafish 1.3.

1. Original method:

   The spike neurons is generated based on a Bernoulli distribution with a firing rate $p = 0.8$ to generate signals with a sparsity similar to the observed data.

$$n \longrightarrow Ca \longrightarrow F$$



(a) (b)

Figure 4.6: a) Signals generated by the synthetic signal generator (image taken from [19]. b) Signals extracted from two neurons of the real dataset. We can see that the real signals and the synthetically generated signals are quite different; the cited paper does not consider bursts of spikes.

Some traces generated by this system can be compared with the traces extracted from the real dataset by looking at Figure 4.6.

2. Burst spiking scheme method:

   This proposal has been made to adapt the synthetic data to the observed data extracted from the real dataset. As we can see in 4.6 some signals in the real dataset looks really like a consequence of a single-spike event, but not others which shape is not exactly an exponential. To generate signals with a behavior similar to the dynamics of the real dataset we have modified the simulation process (using the implementation which was kindly sent to us by Yuriv Mischenko) as follows:

   (a) Generate synthetic signals with sample rate $\Delta r = 1000$ times higher than the capacity of the microscope (r about 10 Hz, i.e., 10Khz).

   (b) Generate a first n_1 sequence as a Bernoulli sequence with a parameter $p = p_{originalMethod}/10000$.

   (c) Combine several n_1 signals to generate relations between them in a set of n_2 sequences.

   (d) Each one of the spikes n_2 is expanded to n_3 by a burst of spikes with:

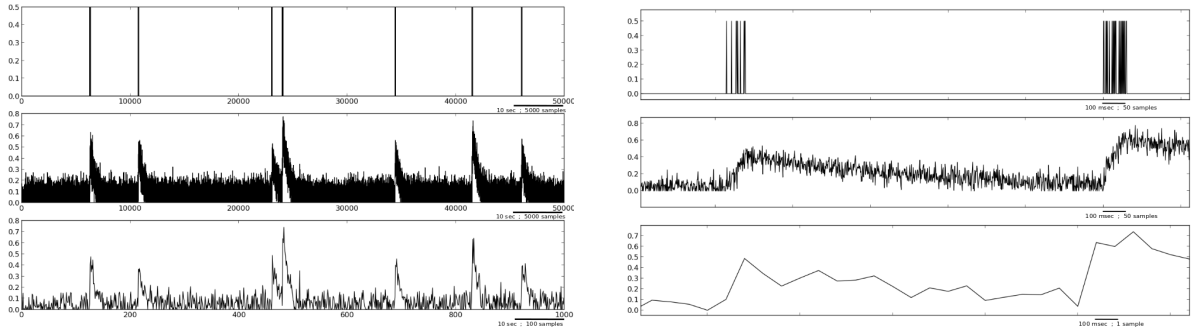
      i. Length uniformly distributed between $\Delta r/40$ and $\Delta r/10$.

      ii. Each time step in these length has a probability of firing defined by a Bernoulli distribution with parameter $p = 0.7$.

   (e) Compute F_2 using n_3.

   (f) F is obtained down-sampling F_2 by a factor Δr. n sequence is obtained by down-sampling n_3. For each sample i the final sequence can be expressed using the expression 4.2.

$$n[i] = max\left(n_3\left[i \cdot \Delta r - \frac{\Delta r}{2}/2, i \cdot \Delta r + \frac{\Delta r}{2}\right]\right) \tag{4.2}$$



(a) (b)

Figure 4.7: This plot show the process we follow to generate a new signal. From top to down, we show the original spike train, the fluorescence signal generated with the classical method and, finally, the last down-sampling. In a) we can see a long trace and in b) we can see a zoom over a specific pair of events in the trace shown in a) where we can appreciate that each event is generated by a hidden burst of spikes.

With this system the signals has a behavior based on the physical properties of the signals we want to analyze and the model is not limited to the single-spike event generated as happens in the original model. The shapes observed can be the result of the addition of the shapes inducted by bursts of spikes and the final observation we can work with is a version of the signal taken sampling at a very low frame-rate (this process is illustrated in the Figure 4.7).

In Figure 4.8 we can see how our signals can reproduce the events which could not be correctly explained with the previous approach. The model proposed is very flexible and is just a generalization of the original one. We expect that when more datasets will be available the parameters of this system can be tunned to generate signals which really fits the real data under analysis.

The last thing we have to model is the relation between different neurons. As we explained in the point 4.2.4 we will use the same approach used in several papers such as [17, 22, 19, 15, 13, 12]. If a set of N spike trains n_i is generated, we use a matrix R of NxN where each element $R_{i,j}$ is used as the parameter p for a Bernoulli distribution which is determines if an spike which appears in n_i will appear in n_j. In figure 4.9 we two pairs of signals with different p values relating them.

The system allows us to generate several cluster schemes and assign different values to each $p_{i,j}$ in R using a uniform distribution between two predefined values depending on if i and j belongs to the same cluster or they not. A representation of matrix R for several configurations can be shown in Figure 4.10.

4.2.2 Event Detector

The implementations of this block take a single signal x with length N representing a fluorescence trace as input. The output y of these blocks must be another array of the same length N.

We will take as a base the work proposed in [19] for modeling the signals and we will understand these "events" as electrical activity in the nerve cell. This electrical activity is understood as a spike in a hidden temporal sequence which generate the observed fluorescence.

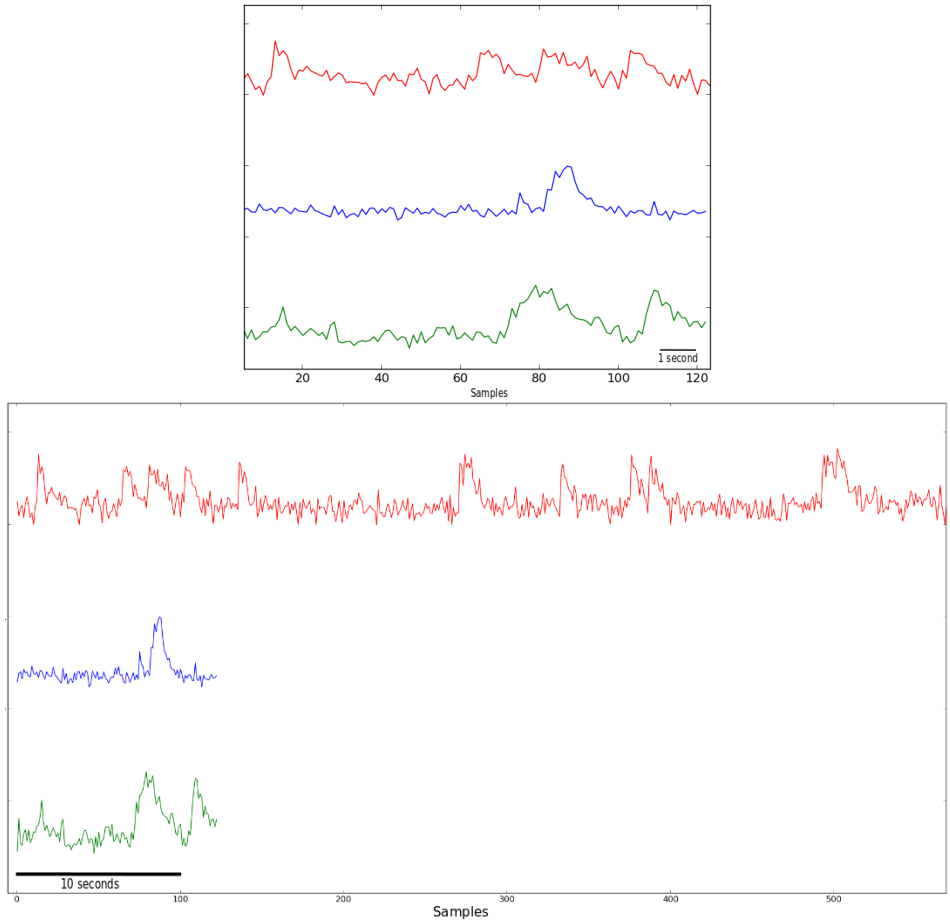


Figure 4.8: Signal generated by the synthetic signal generator "Burst spiking scheme method (red). Signals extracted from detected neurons in the real dataset (blue and green). In the top image we can see in detail these three signals; in the bottom image we see a longer synthetically generated signal trying to show the same dynamics than the real data.

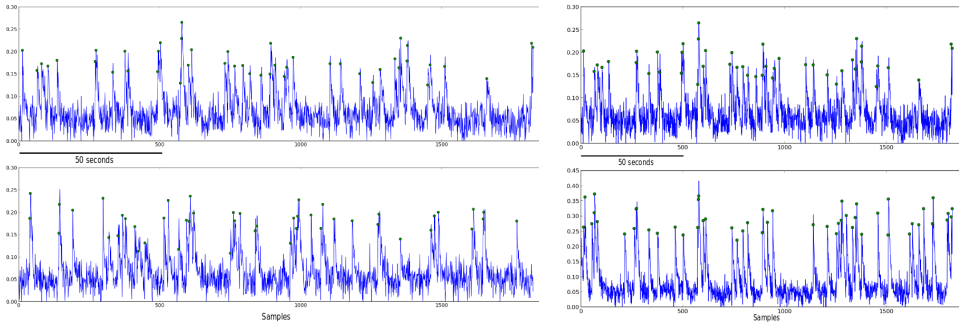


Figure 4.9: Two example of simulated traces for the same period of time. They represent different neurons recorded in the same video using fMCI techniques. In the left we can see an example of non-correlated sequences ($p = 0.05$) and in the right we can see an example of strongly-correlated sequences ($p = 0.8$). Green dots represents the instant when a spike has been generated by our signal generation algorithm. These signals have been generated at high frame rate.

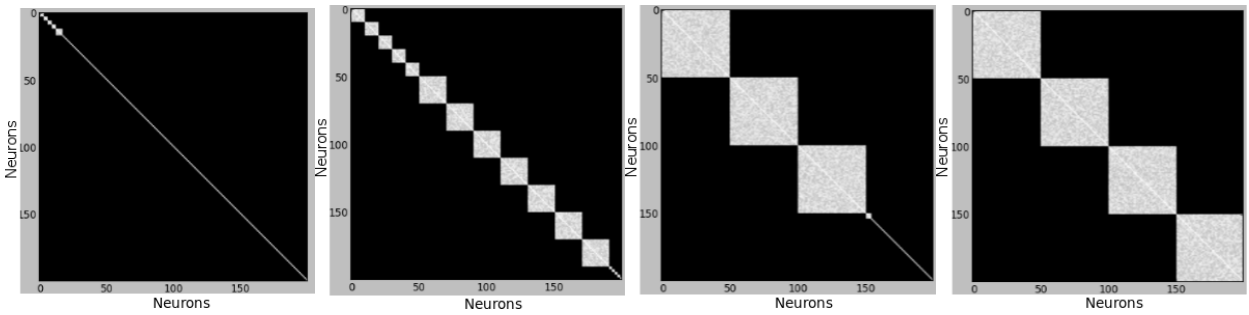


Figure 4.10: Some correlations matrix R which can be easily generated with this synthetic generator module. The first image represents only a few neurons in each cluster and most of them not belonging to any cluster. The second one represents a lot of clusters of different sizes. The third and fourth ones represent big clusters (lots of neurons related). While color means a strong correlation and black weak or non-existing correlation; we can see that our algorithm generates different correlation parameter between each one of the neurons in each cluster (not uniform color in the cluster's representation).

We must remember that this block must do a pre-processing operation before clustering the activity of the detected neurons. As we explained in the Chapter 1 the variability on the observed traces due to the saturation of the cells or their anatomical shape is the reason why the bibliography don't work with the signals directly, but previously finding where the neurons are active or not.

Understanding this spicular activity as events, the output will be a N-length array where each position is:

- The $i-th$ value of y is equal to p if there's an event in x in the $i-th$ position with probability p. Actually, this p value may not be strictly understood as a probability, but its value should be in some way proportional to that probability p.
- The $i-th$ value of y is equal to 0 if no evens have been detected in the $i-th$ position of x .

We have implemented some blocks to implement this events locator. We will explain the implementation of the different implemented blocks being three of them are directly related with the bibliography and being the last one a variation we propose in this project. The limitations of each one of them has been explained in the Chapter 3.1.3, so we won't discuss them again.

1. Monte Carlo technique

   We will use the implementation kindly provided by the author of the paper [19]. The basis of this algorithm were explained in 3.1.3.1. The idea they follow consists on understanding the observation F as the output of a generative process which follows the assumptions explained in the paper.

   Being n the sequence describing the firing of the neuron and F the observations in that statistical model, what they do is, considering F as the observations and n a hidden variable, use a particle filter to find the parameters θ which maximizes the probability of generating F. n is considered one of these parameters. What is finally implemented is the expression 4.3 where an Expectation Maximization (EM) algorithm is used and θ' corresponds to the estimated set of parameters in the previous iteration.

$$\hat{\theta} = \underset{\theta}{argmax} \sum_{t=1}^{T} \sum_{t=1}^{N} \left(P_{\theta'}(H_t^{(i)}, H_{t-1}^{(j)} | O_{1:T}) \quad x \quad ln \quad P_\theta(H_t^{(i)} | H_{t-1}^{(j)}) + \sum_{i=1}^{N} P_{\theta'}(H_t^{(i)} | O_{1:T}) \quad x \quad ln \quad P_\theta(O_t | H_t^{(i)}) \right) \tag{4.3}$$

What we are really interested in is in n, one of these inferred parameters so we want to compute:

$$n = \underset{n_i}{argmax} \left(P\left(n_i | F\right) \right)$$

As we previously commented the paper is very well explained and with lots of details that we are not going to repeat in this report. If the reader in interested in this algorithm, we seriously encourage them to read the original paper [19].

As we discussed in 3.1.3.1 this model is very powerful but very dependent on the input signals and the well fitting of the assumptions made in the generation algorithm with the real data. We are not sure that this method works fine with our real data or even with the synthetic data generated with our proposed "Burst spiking scheme", so we have implemented other methods more flexible and independent on a specific statistical model.

2. Unsupervised adapted filtering approach.

   We implement the same algorithm explained in the point 3.1.3.2. In that point the limitations of this algorithm are discussed.

```
signal = FluorescenceSignal
K_Peaks = 5
filter_length = 5
thrs_acceptance = 0.75

adapted_filter = zeros(filter_length)
filtered_signal = high_pass_filter(0.15, signal)
peak_position = getPeaksPositions(signal, K_Peaks)

for i in length(peak_positions):
    adapted_filter += signal[peak_positions[i]:peak_positions[i]+filter_leng
adapted_filter = Normalize(adapted_filter)

conv = convolve(signal, adapted_filter)
thrs = 0
for i in length(peak_positions):
    thrs += conv[peak_positions[i]]
thrs = thrs / K_peaks * thrs_acceptance
peaks_found = conv*(conv > thrs)
```

3. Exponential Fitting

The algorithm is an approximation of the algorithms explained in the point 3.1.3.3 of this report.

```
signal = FluorescenceSignal
time_up = 4
thrs_up = (max(signal)-min(signal))/2+min(signal)
filtered_signal = high_pass_filter(0.15, signal)
decay_indexs = arange(VALUE_TO_DEFINE, VALUE_TO_DEFINE)
spikes = zeros(length(signal))
for i in length(signal):
    if all(signal[i:i+time_up]>thrs_up):
        amplitude = signal[i]-signal[i-1]
        area_min = area(signal)
        fin_sig = signal
        for j in length(decay_indexs):
            sig_aux = signal - generateExponential(position=i,
                                                   decay = decay_indexs[j],
                                                   amplitude = amplitude
                                                   min_value = signal[i-1])
            area_aux = area(sig_aux)
            if(area_aux < area_min):
                area_min = area_aux
                fin_sig = sig_aux
        signal = fin_sig
        spikes[i] = 1
```

4. Gaussian maximization

We have proposed a variation of the previous algorithm. The two weakest points of the previous algorithm are:

- Very simple method to compute the amplitude of the exponentials. Non valid if the exponential shapes starts with a soft shape. In the theoretical model this should not happen, but in real data sometimes happens.
- Problems with filtering. In literature they use a high-pass filter, but using this filter imply, for example, a final zero mean signal. Taking into account that one of the criteria used in this algorithm is the area and a positive-defined signal, this algorithm won't work fine with data with several events (not zero-mean).

Our idea takes the following assumptions:

(a) As NIG biologists told us, most of the time in a long recording, a given neuron should be non-active.

(b) The bright detected during that time is just a Gaussian random variable ([19]).

We just substitute the high-pass filter by the following method calling x the input signal:

(a) We just define a window of size 20 and compute the kurtosis of each segment.

(b) We set a threshold based on the minimum detected kurtosis in a

(c) We take all the segments with a kurtosis under that threshold as the 'most gaussian' zones in the signal. Assuming that when nerve cells are non active, the bright in non active nerve cells is a Gaussian random variable. We expect these selected segments to be non-active.

(d) The rest rest of the zones are potential active-zones. We substitute these active-zones by a line connecting the previous and posterior non-active zones. Call this signal x'.

(e) Filter x' with a low-pass filter. Call this signal x''.

(f) Build the final signal as:

$$y = x'' + (x - x')$$

We have actually considered another criteria, not only the area to accept an spike. We only accept a candidate as spike if the final area is decreasing and if the kurtosis of that zone decrease after subtracting the candidate exponential.

One of the problems we got is that the strict definition of the kurtosis, when applied over the whole sequence is not good. This is because what we want to check is the gaussianity of the zones where non-exponentials are expected. We have had no time to explore it carefully, but the kurtosis of the whole signal does not follow a continuous decreasing. We tried computing the kurtosis over the filtered signal and then taking a mean equal to 0, but the behavior was neither continuous. Probably it's because while we are deleting events, the part of the signal we are 'cleaning' is more Gaussian than it was, but not necessarily the whole signal. In Figure 4.11 we can see how the described algorithm works and the change in the kurtosis after and before running the described algorithm.

In Figure 4.12 we show an example of how our proposals tend to achieve better results than Monte Carlo method when the algorithm is fed with our simulated data.

4.2.3 Event detector results tester

The objective of this block is to evaluate in a quantitative way the quality of the event detector algorithms. In this project we understand an event as an spike happening due to the electrical activity in a neuron.

The similarity $\mathcal{L}$ between two spike trains n_{ref} and $n_{inferred}$ is computed using the expression 4.4.

$$\mathcal{L} = 1 - \frac{(n_{ref} - \sum n_{ref}) \cdot (n_{inferred} - \sum n_{inferred})}{\sqrt{\|n_{ref} - \sum n_{ref}\|^2}\sqrt{\|n_{inferred} - \sum n_{inferred}\|^2}} \tag{4.4}$$

We discussed that recovering the exact time when an spike has occurred is very difficult and some biologists even think is not possible at all. This exact identification of the time when a spike happens

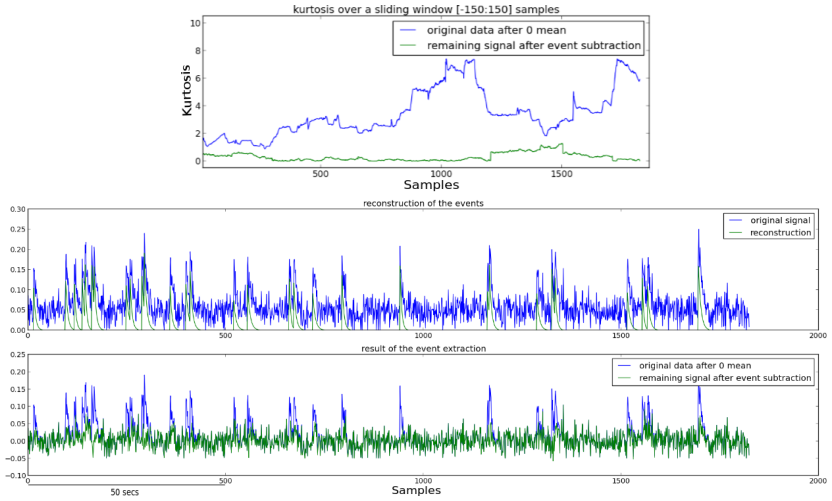
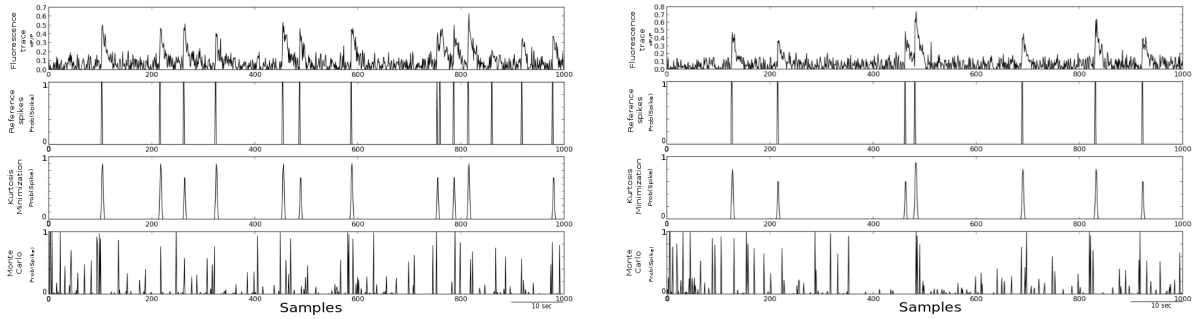


Figure 4.11: In this plot we show the kurtosis computed over a sliding window of 300 samples for a whole sequence before and after detecting the events and subtracting them from the original signal. As we can see, the kurtosis decreases significantly when we use our detector, which means that the data at the end is more Gaussian than it was when the detected events hadn't been deleted yet. In the right plot we show the reconstructed signal (top) and the signal after subtracting the estimated events (bottom).



(a) (b)

Figure 4.12: This plots show some results on two different simulated traces. From top to down, simulated trace, reference spikes (known in generation process), spikes inferred with "Kurtosis Minimization" and spikes inferred using "Monte Carlo". We can see how our algorithm recovers reasonably well the occurrence of the spikes, whilst Monte Carlo is not able to suggest them properly if our data is given.

is even more difficult when we take into account how the spike train n is built in some systems like the system we propose with down-sampling.

Since an error of one or two samples is not a problem when inferring where a spike happens and we are specially interested in a signal which gives us an idea of "something is happening due to the electrical of the neuron", but not necessarily strictly understood as spikes, we propose the convolution of the spicular input signals with a widener-filter with shape $h = \frac{[0.05,0.2,0.5,0.2,0.05]}{\sqrt{\|[0.05,0.2,0.5,0.2,0.05]\|^2}}$. The result can be expressed as shown in equation 4.5.

$$n[k] = \sum_{s \in spikePositions} \delta[k-s]$$

$$y[k] = n[n] * h[n] = \sum_{s \in spikePositions} h[k-s] \tag{4.5}$$

In the realized experiments the results looks more consistent (the values obtained are closer to what a human would qualitatively say) when the 'widening' algorithm is applied both to the reference and the inferred spike train (of course, when we work with synthetic data; as we have discussed if we work with real data we don't have effective tools to test the quality of our results). If this widening algorithm is not used, sometimes appears very high low $\mathcal{L}$ although $n_{inferred}$ and n_{ref} are quite similar.

4.2.4 Signal clustering

Two algorithms has been implemented to clusterize the inputs signals by analyzing the activity of a set of neurons S: meta k-means and agglomerative based.

The input to this module is a set of N signals S and the output is expected to be a NxN matrix R where each position $R_{i,j}$ represents the probability of i and j of being related.

We have some extra information we can use, such as the position of the neurons since some papers assure that is more probable that two neurons which are located close to each other are connected than two neurons which are far to each other.

We will explain the two algorithms we have developed and we will comment the strong and weak points of both of them.

- Meta K-means:

  This algorithm has been implemented as it's described in the bibliography [15] and we discussed in 3.2.

  The problem of this algorithm is that we have to define a parameter k to run the algorithm. The authors of the cited paper assures that a k equal to 3 gives stable results by running the algorithm 1000 times. We agree with them, but we are not sure if this stability is or not good at all. The sensation we have is that this method tends to generate a number of final clusters depending on k. Is quite difficult specially to find clusters with only a few signals if the amount of signals we are working with is big.

  Taking into account that we are expecting to work in the future with thousands of signals and we don't want to impose any previous assumption over the characteristics or size of the clusters,

this difficulty seems an important limitation.

- Agglomerative Clustering:

  To solve the problem which happens when this k-means based algorithm is used, we propose the use of another unsupervised clustering method. We have decided to use an algorithm based on agglomerative clustering methods. The common agglomerative clustering methods share the same algorithmic scheme but differs in the way in which inter-cluster distances are updated after each clustering step. The implementation we use in this paper is called SAHN (sequential, agglomerative, hierarchic, non-overlapping methods) [6].

  In this algorithm we will understand the input signal x_i as the $i-th$ node and define a distance criteria. Then the clustering procedure is as follows:

  1. Let S be the current set of nodes, with implicit or explicit dissimilarity information. Determine a pair of mutually closest points (a,b).
  2. Join a and b into a new node n. Delete a and b from the set of nodes and add n to it.
  3. Output the node labels a and b and their dissimilarity $d(a,b)$.
  4. Update the dissimilarity information by specifying the distance from n to all other nodes.
  5. Repeat steps 1 to 4 until there is a single node left, which contains all the initial nodes.

  We have developed a modification over this algorithm consisting on adding a condition between steps 3 and 4: only consider n as a new node if $d(a,b) \leq \min_{\forall i,j} (d(n_i,n_j)) + \mu_{thrs}$. This method allow us to control how strict we want to be to consider that two signals are closely related.

  This algorithm is very fast and is only based on the distances between each pair of input vectors, so we avoid the problem of choosing a value for k as happens in the k-means based algorithm and the number of clusters are not conditioned by this arbitrarily chosen value.

  This modification, as we will discuss in results chapter allow us to set relations as strong as we want before accept a pair of signals as part of the same cluster and exist the possibility of lots of non-clustered neurons.

  If the input to this algorithm is an estimation of the spike activity, we recommend to use the widening process proposed in the point 4.2.3 to avoid giving a lot of important to the exact location of the inferred spikes.

  Is very practical to use this algorithm with the Results Plotter block, since it's really fast to recompute the results given a non-fixed μ_{thrs} and the results obtained can be displayed in real time.

  The weakness of this algorithm is that is very deterministic and is only based on the distance between the signals and maybe it's a too simple approach to find 'non-obvious' relations between nerve cells.

4.2.5 Clustering results tester

To compute a metric representative on the quality on the clusters obtained we will just compute a metric based on the squared error between each item in both input matrix M_x where each item $M_{x_{i,j}}$ is an scalar which represents the probability of an spike burst in the$i-th$ input signal being propagated in the $j-th$ input signal.

This module only can be used when we use a synthetically generated signal approach, where the information used to relate the different generated spike trains for each neuron is used as a reference M_{ref} and the $M_{inferred}$ from the Signal Clustering block is used as matrix to be compared with M_{ref}.

The result of this block is an scalar $\mathcal{L}$ which represents how similar are the inferred matrix connectivity matrix $M_{inferred}$ and the reference matrix M_{ref} used to generate the synthetic signals.

$$\mathcal{L} = 1 - \sum_{\forall i}\sum_{\forall j} \left(M_{ref_{i,j}} - M_{inferred_{i,j}}\right)^2$$

4.2.6 Results plotter

This block is used to show the results to the NIG in a graphical way. As biologists, they are not interested at all in the previous steps, models or considerations but in the final results. As we explained in the objectives, the NIG biologists want some tools to help them to find relations between neurons in images where hundreds or thousand of neurons are recorded. Their final aim is to use the relations proposed by our system to make some hypothesis based on neurobiology knowledge.

The tool we provide to the NIG takes the output of the Signal Clustering block, a connectivity matrix M where each position $M_{i,j}$ is a binary value which express if neurons i and j are or not related. In case the output of the Signal Clustering block is a matrix M' where each position $M'_{i,j}$ represents the probability of i and j of being related, a threshold μ_{thrs} can be chosen by the users of the visualizer to convert $M'_{i,j}$ in a matrix $M_{i,j}$ suitable to be plotted as expressed in equation 4.6.

$$M_{i,j} = \begin{cases} 1 & \text{if } M'_{i,j} > \mu_{thrs} \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

The others inputs required are a representative frame of the dataset analyzed and for each neuron a pair of coordinates.

This module can only be used with the real data analysis scheme. If in future development someone is interested on developing a image synthetic signal generator such as the one described in [31], this block could be used.

In Figure 4.13 an example of the output delivered to the NIG users is shown.

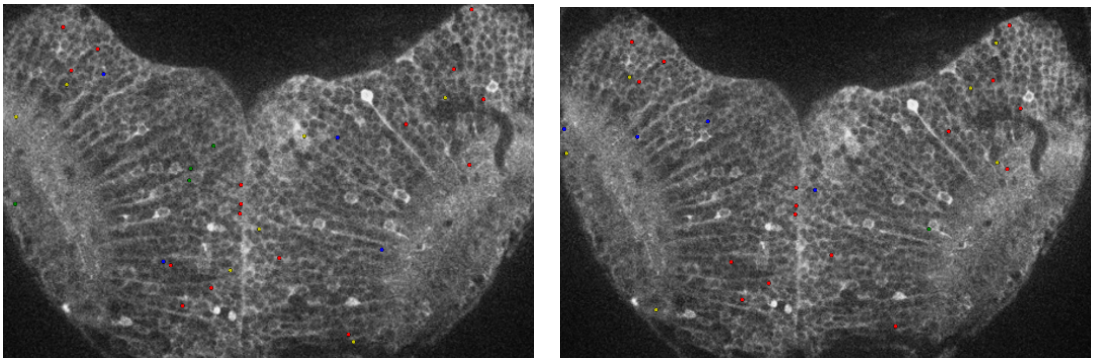


Figure 4.13: Connectivity matrix display results using the Results Plotter block set with different thresholds.

Chapter 5

Results

In this chapter we will try the algorithms developed and which were explained in the Chapter 4. Our final goal is to test these algorithms in order to estimate our capacity to suggest valuable hints to the NIG biologists based on the relation between the electrical activity of the neurons observed.

The main problem to reach this final goal is:

- There's no information provided which could be used to check if our results can give these interesting hints we are trying to find.
- The dataset we have is very short and some of the algorithms proposed could not work properly due to this shortness.

Due to these limitations, it's very difficult to evaluate which of the proposed algorithms and methods are promising and should be improved and which ones should be discarded. Actually, the second limitation is only a problem for this project, but in the future, when new datasets will be provided by the NIG, that problem should disappear. Despite the fact that the second exposed limitation is only an eventual limitation, the first one won't be solved with more data.

As we explained in previous chapters, is because of this reason we have developed a system to generate synthetic signals with a similar behavior to the available dataset.

What we will do in this chapter is

1. Analyze the behavior of each one of the systems proposed using synthetic data and we will try to extract some conclusions of which systems are promising and which ones are not.
2. Once we will know how each one of our algorithms works with synthetic data, we will try the system which seems the most promising and we will launch them with our real dataset. The evaluation of this second part will be purely qualitative. This results will be sent to the NIG biologists and we expect other researchers will continue developing this framework to adjust it to the NIG feedback.

5.1 Results using synthetic data

In previous chapters we explained the model we are using to relate the different neuron's activities. The final aim of the project is to recover some information of this relations by observing neuron's activity, so we will set our tests on this final objective.

We will start explaining the generation schemes we will follow and how the evaluation of these systems will be done. After that we will present some systems and the block combination required to launch each one of these systems and we will define a nomenclature to refer to each one of these systems. Then we will show the results achieved with each one of these systems and we will conclude this point commenting these results and trying to give an interpretation to the obtained results.

5.1.1 Dataset generation

First of all, we will test our systems with the scheme proposed by Volgenstein et. al. We will denote this system as "VMC" ("Volgenstein Monte Carlo").

The expected results using VMC generators are very good, but we are actually interested in our synthetic generation system using bursts of spikes. We denote this system as "BGS" (Burst Generation System). As we explained in 4.2.1.2 this systems is based on a VMC generation scheme using a very high sample rate and then we down-sample this signal to make them similar similar to the real data our system is supposed to analyze.

We want to check the behavior of these algorithms using different amounts of signals to test the scalability of the system and the capacity to work with clusters of different size. We will name a system with N total signals as follows:

$$"BGS - N" + \sum" - i, \#j"$$

where the summation is used to express concatenation of strings using a "-" to separate the new sub-word of the general one and j represents the quantity of clusters with i items.

For example, a dataset with: 20 signals, 3 clusters of 3 signals each one, 2 clusters of 4 signals each one and 3 independent signals would be named "$BGS - 20 - 3, 3 - 4, 2$".

We propose some sets of tests:

1. Same-sized clusters and no unclustered signals:
    - BGS-12-3,4
    - BGS-20-5,4
    - BGS-200-50,4
2. Non-same-sized clusters and no unclustered signals:
    - BGS-12-2,3-3,2
    - BGS-20-2,4-3,4
    - BGS-200-10,5-20,4-20,3-2,5

3. Non-same-sized clusters and unclustered signals

    - BGS-200-50,3-4,1

4. Small clusters in a mostly non clusterized space:

    - BGS-200-3,4-5,1

5.1.2 Systems tested

We will test all the combinations of clustering algorithms and spike detector algorithms we have implemented. We will add a new block to the spike detector; a block which just take the labels generated by the synthetic data generator and return them as a result. In this case, the performance of the intermediate step consisting on detecting the events will show a perfect performance.

This will allow us to evaluate exclusively the clustering algorithm in presence of no errors in the event detection and we will be able to estimate the impact of the errors in the detection on the final clustering performance.

The nomenclature used will be as follows:

- Clustering
    - K-means based clustering$\longrightarrow$KC
    - Hierarchical clustering $\longrightarrow$HC
    - Hierarchical clustering (μ_{thrs}=0.7)$\longrightarrow$HCT
- Spike detector
    - Monte Carlo detector $\longrightarrow$MCD
    - Unsupervised adapted filtering$\longrightarrow$AFD
    - Exponential fitting $\longrightarrow$EFD
    - Exponential fitting using Kurtosis to extract the baseline$\longrightarrow$KEFD
    - Fake detector returning generation labels $\longrightarrow$FD

5.1.3 Results format

As the signals are generated as random processes we consider that a good way to test each method is launching the same method over several realizations of the same random process. We will show the mean of the results achieved with sets of 10 experiments for each method and generation process.

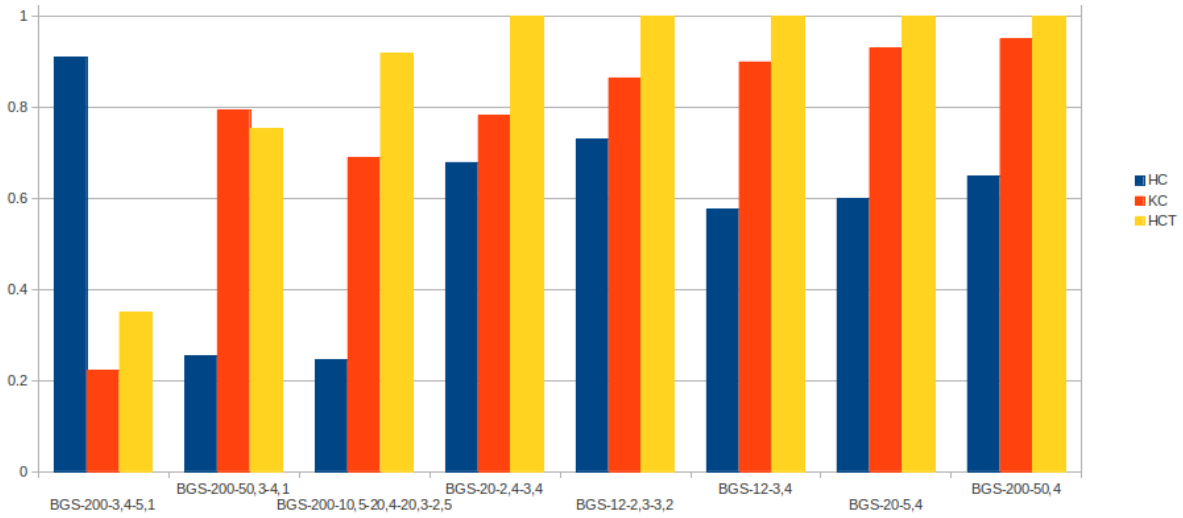


Figure 5.1: Results obtained applying the different proposed clustering techniques to the proposed datasets.

5.1.4 Testing

5.1.4.1 Clustering algorithms testing:

We will test how our clustering algorithms work under each proposed hypothesis. To do it we want to avoid uncontrolled variables as the performance of the spike detector algorithm. Because of that, we will use in all cases the FD detector.

In Figure 5.1 we show the results obtained by using the different clustering algorithms. As we discussed in the proper chapter, K-means show a very good behavior when there are big and dense clusters but it does not work so good when there are non-clustered signals or the existing clusters are small.

In our implementation and following the suggestions made in the bibliography we chose a value of K equal to 3. It may be that using another value this small clusters are easier to be found but the fact that a K value must be defined will still be a limitation for these techniques when they it will be used with real datasets where we won't have any idea of the amount of existing clusters.

Regarding to HC and HCT, it's important to understand the output of each one of these methods. The output of HC for a pair of vectors i and j represents the value assigned by this algorithm to the relation between these two signals (between 0 and 1). The output of HCT is just a binary value, assigning 1 to the relation between i and j if the output in HC for this relation is higher than a given threshold and 0 if it isn't.

We can see how the behavior of HCT is very similar to the K-means algorithms (generally a bit better). It's important to understand that HC seems very bad compared with the other two, but that is not important; what is happening is the reference uses a binary value for each relation and HC a continuous value, so the difference between two results seems very big, although this difference is not significant (when we use a threshold, we can see that the results are good).

What is very interesting is what happen in the dataset with only a few signals actually related and

the rest generated in an absolutely independent way. In this situation, HC is useful to get a very good intuition of what's happening and what is the real relation between the signals in this non-related dataset. In this case, the selected threshold doesn't give us good results and neither k-means does.

This results show how there's not a 'perfect' algorithm to find the underlying relation between signals and the performance of the algorithm will be very dependent on the unknown generation scheme. We should discuss this carefully with NIG scientists and try to know more about the expected relation between the analyzed neurons.

If they expect to find big clusters, we should probably use a k-means based clustering algorithm. If they feel that we should only find relations between a few neurons, we should probably choose a scheme based on hierarchical clustering.

Although the results looks quite good, we must be very critical and understand that they are working fine to find relations given the proposed scheme we have decided to work with. The idea of this scheme is to generate signals with a behavior which look similar to the real data but we are still doing some assumptions which are probably not correct and which probably don't fit with the real structure of behavior of the fishes' brains. The most important limitation of our systems is we are considering an stochastic connection model where we only obtain a value saying if the neurons are or not connected. In real brains, it seems that different kind of activity exist and, although some sets of neurons are connected following a model similar to how we presented in this project, weaker connections exist connecting these "clusters of neurons".

The exploration of a different approach capable to work with these kind of connection structure should be explored in future developments taking into account the information provided by NIG biologists.

5.1.4.2 Impact of spike detection performance in clustering performance

In the previous point we used the information we got in the generation process to evaluate the performance of our clustering algorithms under perfect speak detection scenario.

We will now evaluate the impact of a non-perfect inference of spikes. This is very important because as we have discussed in previous chapters, in real data is very probable to not find spike locations. We will only find zones of probable activity. We do really want to study if our clustering algorithms will work properly if the spike detection algorithms don't work fine at all.

In a first step we will discuss the performance of each of the algorithms for spike detection and we will discuss the effect of the "widening" we use to avoid problems regarding to the exact localization of each spike.

Once this two results will have been discussed, we will show the results of the final clustering algorithms and their relation with the chosen spike detector.

In Figure 5.2 we show the performance of each one of the spike detection algorithms. As we can see, Monte Carlo methods are not working properly with our generation method. This result was expected and discussed in the proper point. The particle filters in which Monte Carlo Detector method is based is very dependent on the generation method and in the statistical characterization of this method. In our case, where this generative model is different, this particle filter don't achieve good results.

Regarding to the rest of implemented detectors, all of them seem to work quite well. KEFD looks

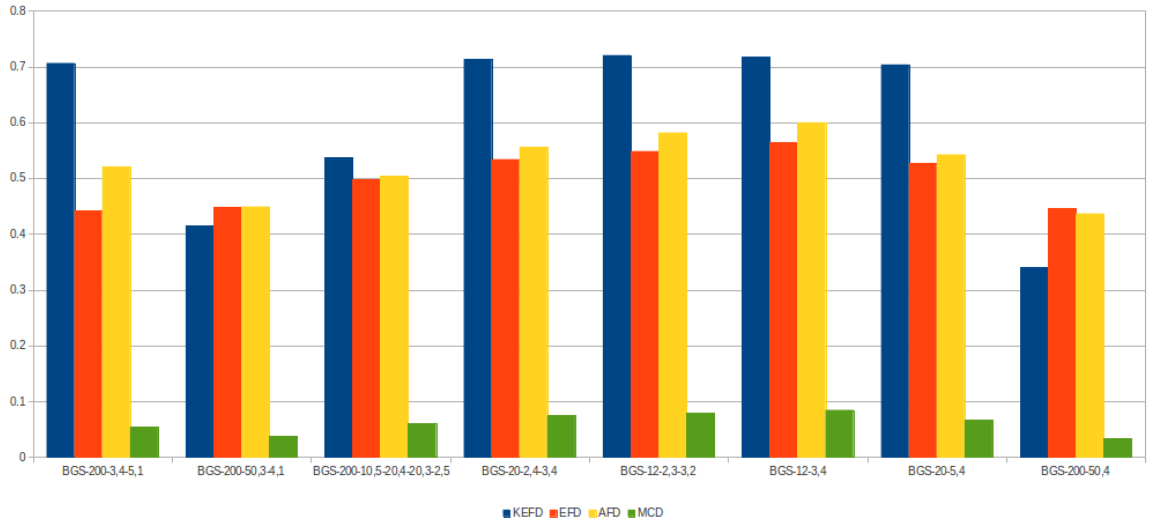


Figure 5.2: Performance of each one of the implemented spike detectors over the proposed datasets.

better in some datasets, but these results are not significant at all. The proposal for a good detector must be done with the real dataset but at least we can feel confident with the behavior and the generality of the implemented algorithms.

Although some of these algorithms seems to be more interesting to recover the hidden spike events, we must remember that the detection of those spikes is not our final goal. These spikes are a formalism which allows us to understand the signals we are working with and help us to link our techniques to process the signals with the biological events we are trying to know more about.

What is actually interesting is to know if, despite a bad detection, we can get a good idea of the existing correlation between the different involved signals. In Figures 5.3 and 5.4 we show some graphics which shows the results achieved when we try to find clusters using different spike detectors.

The feeling we get from these Figures is that the results in clustering are quite independent on the results in the detection of spikes. We can see how, when the detection is very bad such in Monte Carlo Detector, the results in clustering a very bad in comparison with the rest of detectors. In the rest of situations, when the detector achieves a performance of 0.6-0.7 the results in clustering are not so different of the achieved results when the reference spikes are used as the output of a hypothetically perfect detector.

The last thing we discuss is the use of a widening algorithm. The idea of these algorithms is not to give some flexibility to our algorithms regarding to the exact position of a detected spike.

In the Figure 5.5 we show the results in clustering when we use a widening algorithm and when we don't use it. We can see how the results are better when we use a widening algorithm.

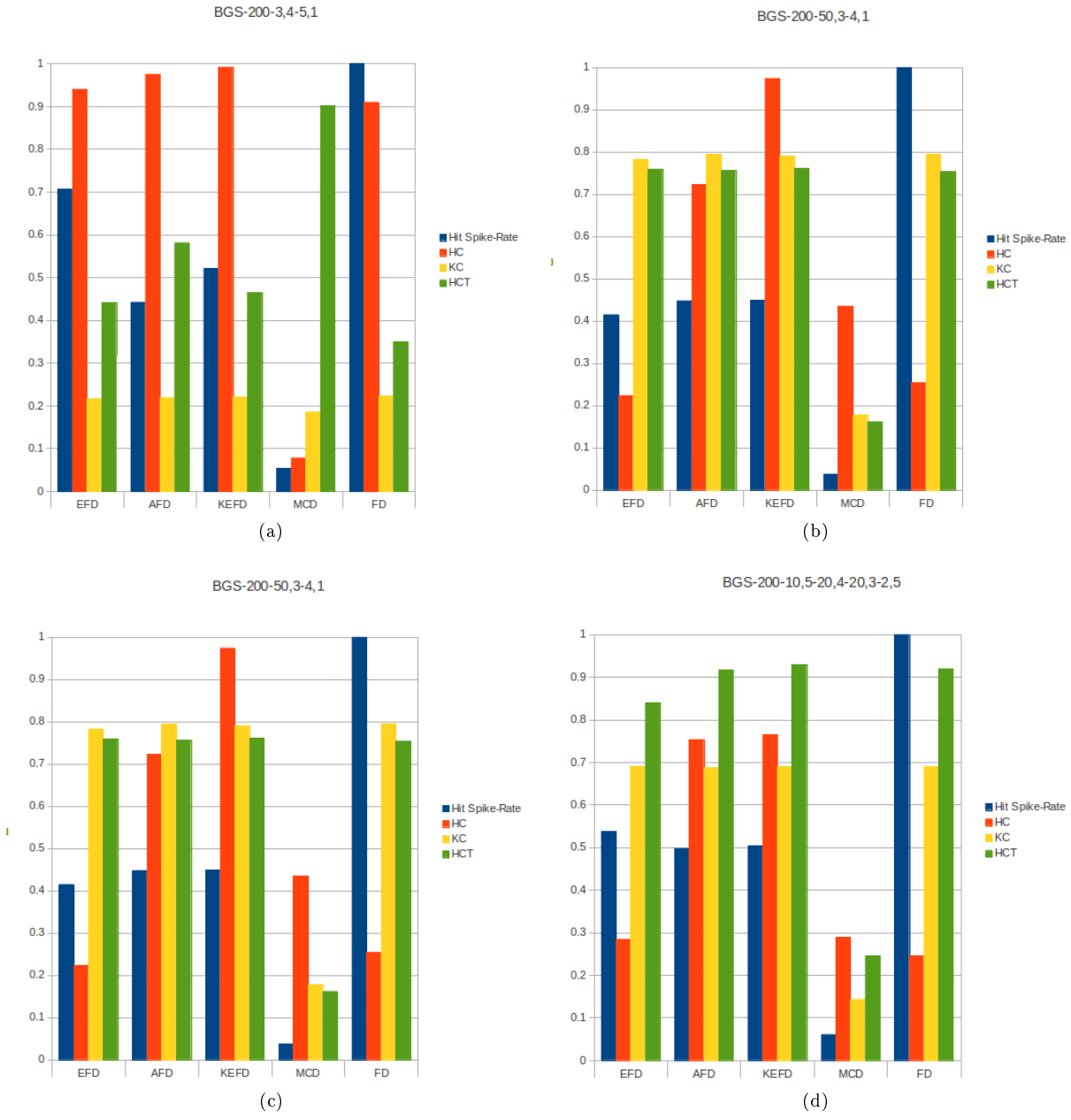


Figure 5.3: For each dataset, we plot the results obtained in speak detection and in the final clustering using that speak detection algorithm.

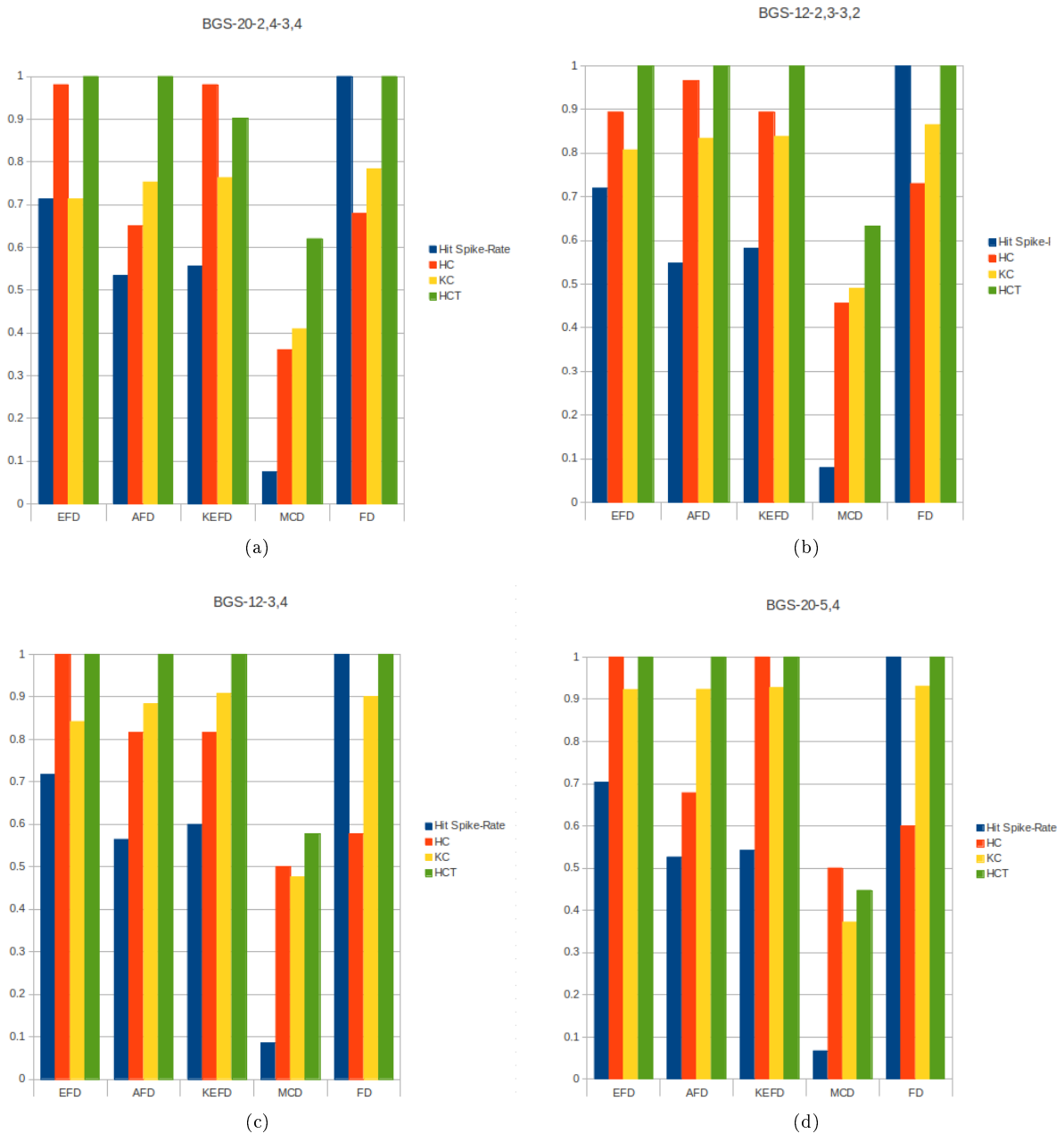
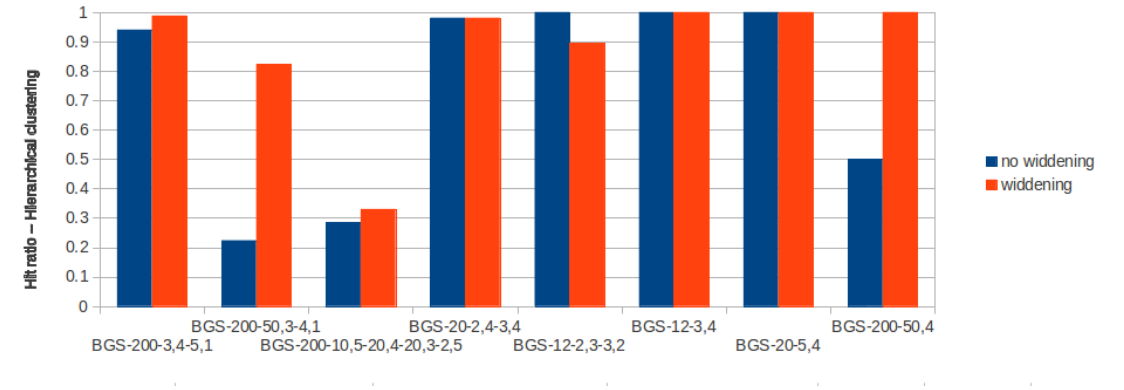
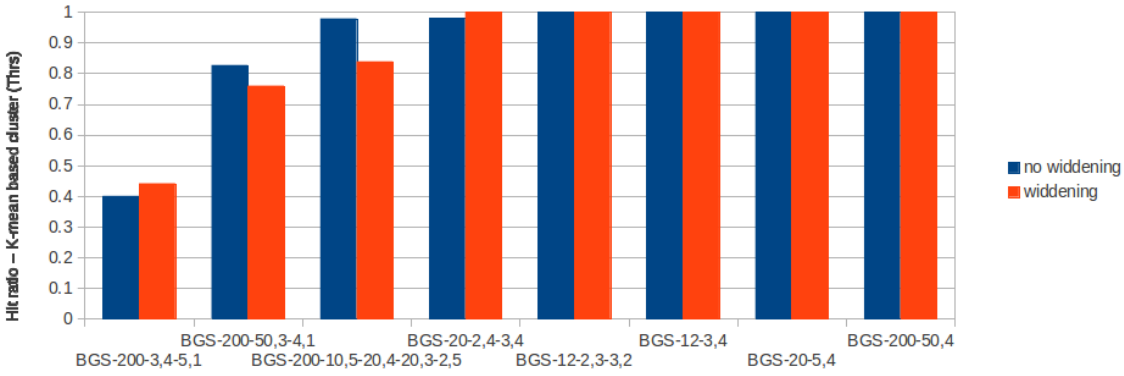


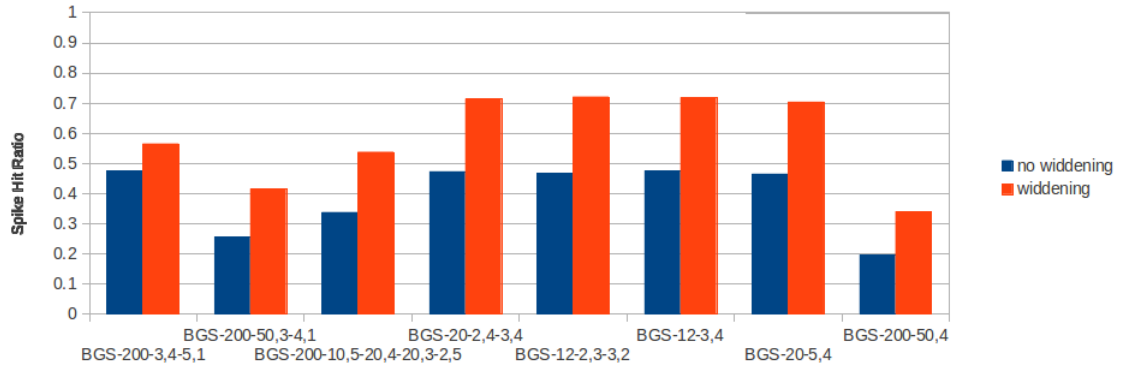
Figure 5.4: For each dataset, we plot the results obtained in speak detection and in the final clustering using that speak detection algorithm.



(a)



(b)



(c)

Figure 5.5: For each dataset, the results using each one of the clustering algorithms with a widening algorithm and without it.

This results are quite promising, because they give us an idea of a reasonably good recovering of the relation between the different signals even though we don't reach a perfect detection of the spikes and this is precisely the situation we expect when we will work with real data. Even though we can recover the clusters without an exact detection, the use of a widening algorithm increase the final performance of these clustering algorithms. This probably means that, although the exact detection of all the spikes is not necessary, it's important to develop algorithms which give us an idea of neuron activity is happening (not so necessary to know the exact time when this activity happens).

5.2 Results using real data

In this point we will test the most promising set of blocks with our real data. As we explained in the introduction of this chapter, we will only show the potential to use our framework to analyze these kind of signals. The results will be evaluated just in a qualitative way expecting to receive feedback from the NIG in the future.

We start analyzing the only available video. We locate the neurons on this video and extract the bright corresponding to the area of these neurons by using the procedure described in the point 4.2.1.1 of this report. As we explained in that point, the algorithm to detect nerve cells is still under development and in the current state of development we are working with hundreds of False Positives (i.e. points detected as neurons which are not actually neurons). We take the K points identified by this previous version of the neuron locator algorithm.

We take all these neurons and start running the described pipeline using a configuration based on KEFD detector and agglomerative hierarchical clustering. Then we use the block to plot the results obtaining the results shown in Figure 5.9. In that Figure we can see 4 of the clusters we get when we run the algorithm using different threshold values. As we can see in the Figure 5.6 the algorithm seems to clusterize together signals which are actually similar (but we must remember we previously find spikes in the signal, so if the spike detector does not work well, the clustering won't).

As we can see, there is a cluster which is surprisingly stable even if we modify the value of the threshold. The other clusters, when we observe the video have a similar behavior, but are not so clear as this (red) cluster. One of the limitations we can see in this images is the actual neuron detection; lots of False Positives are recorded and is difficult to our algorithm to distinguish the noisy data to the event with real patterns.

In Figure 5.8 we have run the same algorithm using meta-k-means for clustering. The result is non meaningful and some signals are clustered together randomly (even with a huge quantity of iterations, which should stabilize the result) and neurons and noisy signals are clustered together. Because of that and observing the available dataset, agglomerative clustering seems to be an algorithm more powerful to be used with these kind of signals than meta-k-means.

We have observed that the use of the spike detector as a preprocessing step is very useful to discard lots of False Positive neurons since no neuronal activity is found by the spike detector. Nevertheless, it would be interesting to check the behavior of this algorithm with more accurately selected/located neuron candidates. In Figure 5.9 we can see the estimated correlation matrix of the candidates. As we can see the matrix is very big; more than it should taking into account the amount of visible nerve

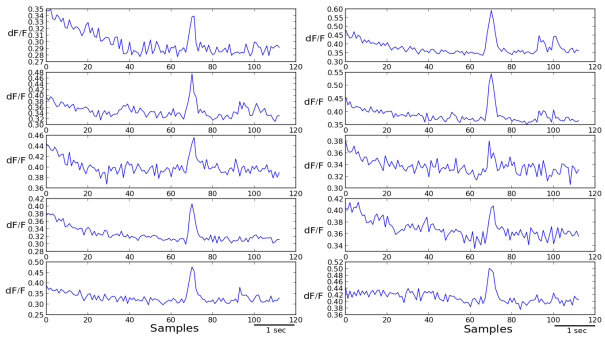


Figure 5.6: Example of 10 signals extracted from the available dataset. Each signal corresponds to a different neuron. The five neurons in each column has been clustered in the same cluster.

cells in the recorded video. Most of these points are just False Positives.

The most important and promising conclusion would be the fact that we can extract some very stable clusters independently of the chosen value for the threshold and we can be less selective if we relax the value of that threshold. This is an absolutely desirable behavior since the NIG biologists can start setting a very low threshold and analyzing the most promising clusters and then increase the value of that cluster to analyze connections which are not so clear.

If we take some random neurons and analyze the performance of the spike detectors algorithms we can see how the detection not always looks good or meaningful (Figure 5.10). That spike locator work well with our synthetic data but not at all with this real data. That probably means that our model to generate synthetic data is not as good as it could be. The point is that now, with the sort data we have, we can't do much to infer the proper parameters to model a signal like the reference: the reference is not long enough to do it. As discussed checking the results objectively is not possible with real data, so the best choice would be to wait for new data to be available and then, modify or generate another synthetic data generator whose signals fits with the signals we want to analyze. Once we have this generator, we could try to improve our spike detectors or maybe choose a completely different approach.

We must remember that there's a lots of assumptions done to build this framework. Since this is a field which is under study and no much is known about the brain and the secrets it hides, these assumptions can be true or not and nowadays there's still no way to know it. No matter what NIG biologist would say about the obtained results, we should not forget the strongest assumption we have imposed: a model for the connectivity of the matrices (we define, too, a model to the generation of each individual signal, but this point is well documented and we have taken a very general approach, so the capabilities of the algorithm are probably not being limited); even if the feedback of NIG is positive, we should always be critical of the models developed and the made assumptions.

We can't actually say much more about how our system works with real data but, as we said previously, we expect that NIG biologists will give future researchers some feedback of these results to focus the next steps of this research in the proper direction to provide NIG biologists useful techniques to continue contributing to the understanding of the brain.

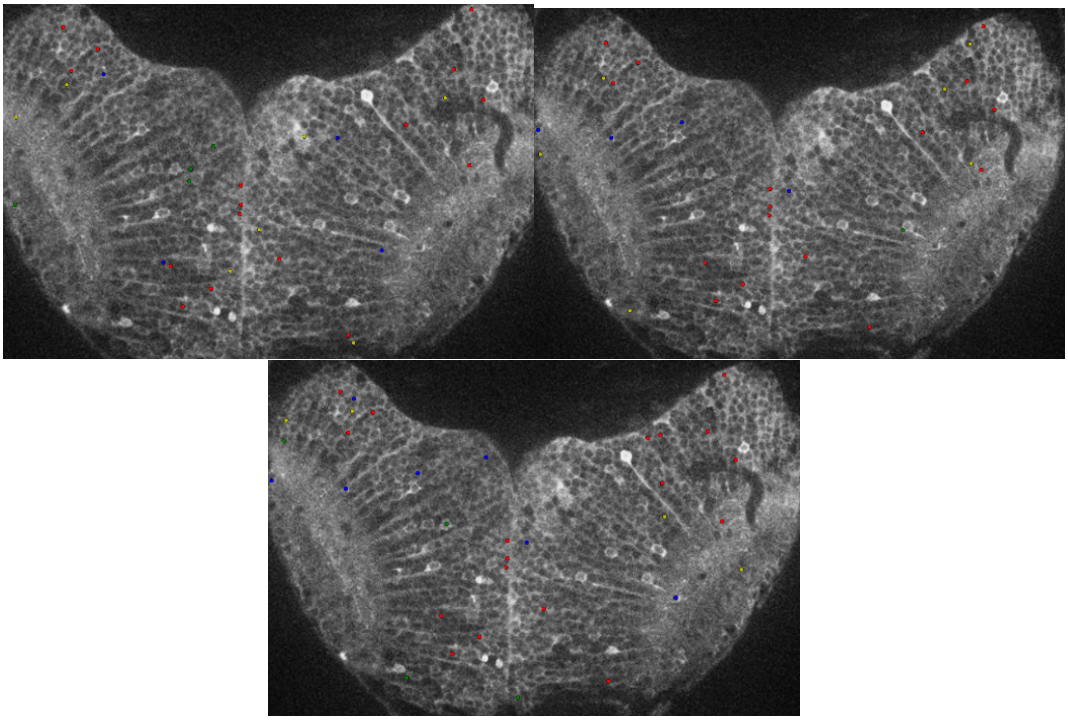


Figure 5.7: Results of clustering neurons in real data using different values for the threshold (left to right: 1.05, 2, 10).

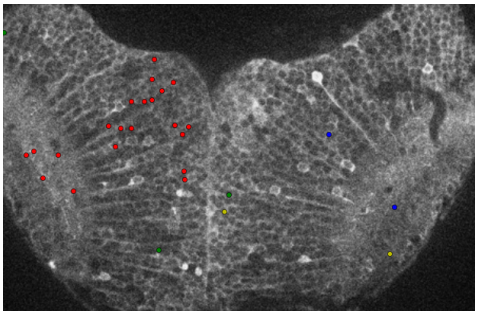


Figure 5.8: Algorithm running meta-k means algorithm. We can see how it tends to build big clusters, even if such a big cluster is non meaningful at all. This meta k-means algorithms tends to mix neurons with noise.

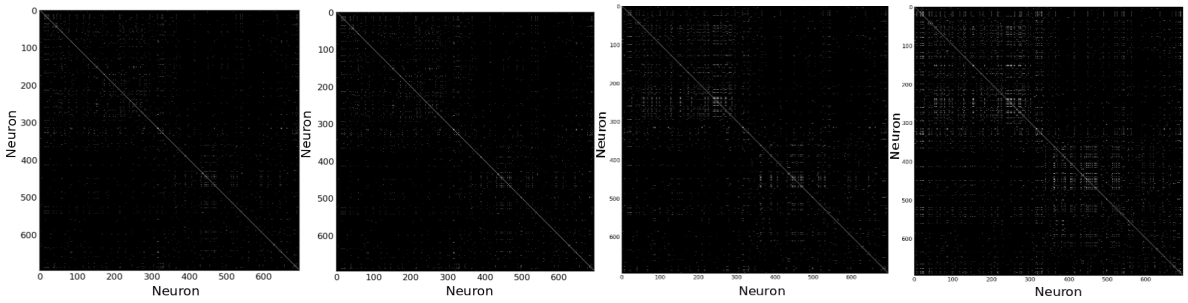


Figure 5.9: Results of clustering neurons in real data using different values for the threshold (left to right: 1.05, 1.1, 2, 10).

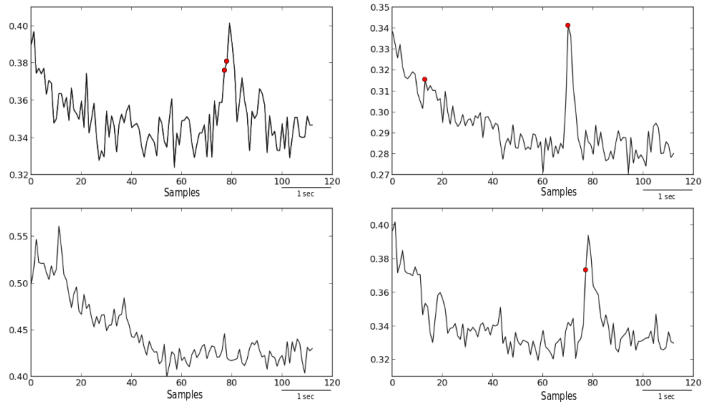


Figure 5.10: Results of spike detection in real data (the red point represents the detection of a underlying spike). The detector detect some events which looks clearly as neuron spicular activity consequence, but probably it lose some shapes we, as humans, would identify as spike.

Chapter 6

Conclusions

In this project we have discussed some of the main bibliography related with neuronal activity analysis obtained with fMCI techniques. We have developed and successfully tested these algorithms with synthetic data using a synthetic data generator implemented based on the bibliography to make the generated signals as similar as possible to the available data.

Although these algorithms worked quite well when they were used with synthetic data, the small size of the available dataset makes unfeasible the test of these algorithms with real data.

Even with synthetic data we can't assure that our algorithms works well. We have implemented and proposed some improvements on the algorithms related in the bibliography to make them work properly with the generated data but a lot of questions are still unanswered in the assumptions made to develop the synthetic explained model (such a probably not sophisticated enough neuron connectivity model). The developed algorithms work well with the synthetically generated data, but due to the lack of external information (such as labeling or annotations) and the lack of a bigger dataset we can't assure that the synthetic model fits well with the expected real data.

Because of these reasons we can't conclude that our framework is useful to properly analyze this kind of questions. Future work may start by sharing the obtained results when real data is used with the NIG and carefully analyzing the received feedback to tune the assumptions made (or even to reject some of the proposed methods).

Independently of the NIG feedback or the correctness of the made assumptions in this report, the most important problem which should be solved is the lack of a bigger dataset. Most of the problems faced during the development of this project should have been solved with more data. A more complete dataset should help to solve problems like the generalization of the algorithms when used with different fishes and when these fishes are exposed to different stimuli or the fine tunning of parameters when parametric models are proposed. Some well defined labels or annotations on the provided dataset may help a lot to the training ant testing of Machine Learning algorithms.

Bibliography

[1] A Bleau and L. Leon, "Watershed-based segmentation and region merging," Comput.Vis. Image Understand., vol. 77 no. 3, 2000, pp. 317-370

[2] Abdullah Mueen, Eamonn Keogh, Qiang Zhu, Sydney Cash, Brandon Westover, "Exact Discovery of Time Series Motifs".

[3] Arnaud Doucet, nando de Freitas, Neil Gordon, "An Introduction to Sequential Monte Carlo Methods"

[4] Arnaud Doucet, Adam M. Johanset, "A tutorial on Particle Filtering and Smoothing: Fifteen years later"

[5] Benjamin F Grewe. Dominik Langer, Hansjörg Kasper, Björn M Kampa & Fritjof Helmchen, "High-speed in vivo valcium imaging reveals neuronal network activity with near-millisecond precision". Nature America, Inc. 2010.

[6] Daniel Müllner, "fastcluster: Fast Hierarchical, Agglomerative, Clustering Routines for R and Python". Journal of Statistical Software, May 2013, Volume 53, Issue 9.

[7] Diana Smetters, Ania Majewska and Rafael yuste, "Detecting Action Potentials in Neuronal Populations with Calcium Imaging". Academic Press 1999.

[8] "Discriminative Didctionary Learning with pairwise Constraints (DDL-PC)"

[9] EranFarida Zehraoui and Younès Bennani, "M-SOM: Matricial Self Organizing Map for sequences clustering and classification".

[10] Eva L. Dyer, Marco F. Duarte, Don H. Johnson, and Richard G.Baraniuk, "Recovering Spikes from Noisy Neuronal Calcium Signals via Structured Sparse Approximation".

[11] Eran A. Mukamel, Axel Nimmerjahn and Mark J. Schnitzer, "Automated Analysis of Cellular Signals from Large-Scale Calcium Imaging Data", Neuron 63, September 2009, pp 747-760.

[12] Feber le, Joost and Rutten, Wim and Stegenga, Jan and Wolters, Pieter and Ramakers, Ger and Pelt van, Jaap, "Cultured cortical networks described by conditional firing probabilities". Conference Proceedings of the 5th International Meeting on Substrate Integrated Micro-Electrode, 2006.

[13] G. Xiong, X.Zhou, L. Ji, "Automated segmentation of drosophila RNAi fluorescence cellullar images using using deformable models," IEEE Trans. Circ. Syst. I, vol. 53, no 11, pp. 2415-2424, Nov. 2006

[14] G. Xiong, X.Zhou, L. Ji, P. Bradley, N. Perrimon, and S. Wong, "Segmentation of drosophila RNAI fluorescence images using level sets," in Proc. IEEE Int. Conf. Image Process., 2006, pp. 73-76

[15] Ilker Ozden, H. Megan Lee, Megan R. Sullivan and Samuel S.-H. Wang, "Identification and Clustering of Event Patterns From in Vivo Multiphoton Optical Recordings of Neuronal Ensembles", Journal of Neurophysiology,July 2008.

[16] Irini Reljin, Branimir Reljin, Gordana Jovanovic. "Clustering and Mapping Spatial-Temporal Datasets Using SOM Neural Networks", Journal of Automatic Control, Vol 13, 2003.

[17] J.Arias, R. André-Obrecht, J. Farinas, "Unsupervised signal segmentation based on temporal spectral clustering". 16th European Signal Processing Conference, 2008.

[18] Jessica Lin, Eamonn Keogh, Li Wei, Stefano Lonardi, "Experiencing SAX: a novel symbolic representation of time series". Data minnimg knowledge discovery, 2007.

[19] Joshua T. Volgelstein, Brendon O. Watson, Adam M. Packer, Rafael Yuste, Bruno Jedynak and Liam Paninski, "Spike Inference from Calcium Imaging Using Sequential Monte Carlo Methods". Biophysical Journal, Volume 97, July 2009, pp. 636-655.

[20] Kazuyuki Samejima, Kenji Doya, Yasumasa Ueda, Minoru Kimura, "Estimating Internal Variables and Parameters of a Learning Agent by a Particle Filter"

[21] Laurent Moreaux, and Gilles Laurent, "Estimating firing rates from calcium signals in locust projection neurons in vivo". Frontiers in neural circuits, November 2007.

[22] Liliana A. S. Medina and Ana L. N. Fred, "Clustering Data with Temporal Evolution: Application to Electrophysiological Signals". Springer, 2011.

[23] M. Baccar, L. A. Gee, R. C. Gonzalez, and M. A. Abidi, "Segmentation of Range Images via Data Fusion and Morphological Watersheds," Pattern Recognition, 29(10), October 1996, pp. 1673-1687.

[24] Naoya Takahashi, Shigeyuki Oba, Naoto Yukinawa, Sakiko Ujita, Mika Mizunuma, Norio Matsuki, Shin Ishii and Yuji Ikegaya, "High-Speed multi-neuron Imaging Using Nipkow-Type Confocal Microscopy". Current Protocols in Neuroscience, October 2011.

[25] Naoya Takahashi, Takuya Sasaki, Atsushi Usami, Norio Matsuki, Yuki Ikegaya, "Watching neuronal circuit dynamics through funcional multi-neuron calcium imaging (fMCI). Neuroscience Research, 2007-

[26] Ogmacop Ramirez, Pablo Sprechmann, and Guillermo Sapiro, "Classification and Clustering via Dictionary Learning with Structured Incoherence and Shared Features". IEEE, 2010.

[27] Q.Chen, X. Yang and E. Petriu, "Watershed segmentation for binary images with different distance transforms," in Proc. HAVE 2004 – IEEE Intl. Workshop on Haptic, Audio and Visual Environments and their Applications, Oct. 2004, pp. 111-116.

[28] Ramon y Cajal, "La textura del Sistema Nerviosa del Hombre y los Vertebrados". 1904

[29] Ramon y Cajal. "Recuerdos de mi vida: Historia de mi labor científica". 1923.

[30] R. Gonzales and R. Woods, "Digital Image Processing", Prentice Hall 2002.

[31] Ryuichi Maruyama, Kazuma Maeda, Hiroyoshi Miyakawa, Toru Aonishi, "Detection of cells from calcium imaging data by non-negative matrix factorization".

[32] Shu Kong and Donghui Wang, "A Dictionary learning Approach for Classification: Separating the Particularity and the Commonality".

[33] Simon J. Godsill, Arnaud Doucet, and Mike West, "Monte Carlo Smoothing for Nonlinear Time Series"

[34] T. R. Jones, A. Carpenter, and P. Golland, "Voronoi-based segmentation of cells on image manifolds", in ICCV Workshop Comput. Vis. Biomed. Image Appl. (CVBIA), 2005 pp. 535-543

[35] V.Luc and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," IEEE Trans. Pattern Anal. Mach. Intell., vol. 13, no. 6, 1991, pp. 583-598.

[36] Yuan-Li, Jessica Lin, Tim Oates, "Visualizing Variable-Length Time Series Motifs".

[37] Yuki Ikegaya, Gloster Aaron, Rosa Cossart, Dmitriy Aronov, Ilan Lampl, David Ferster, Rafael Yuste. "Synfire Chains and Cortical Songs: Temporal Modules of Cortical Activity". SCIENCE, vol 304, April 2004.

[38] Yuriv Mischencko, Joshua T. Vogelstein and Liam Paninski, "A Bayesian Approach for Inferring Neuronal Connectivity From Calcium Fluorescent Imaging Data". Annals of Applied Statistics.

[39] http://www.neuro.uoregon.edu/k12/FAQs.html